

IBM z/OS V2R2: Availability Management

Keith Winnard

Wilson de Figueiredo

Redelf Janßen

Paulo Cesar Nascimento

Ewerton Waki





International Technical Support Organization

IBM z/OS V2R2: Availability Management

December 2015

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (December 2015)

This edition applies to Version 2, Release 2, of IBM z/OS (5650-ZOS).

© Copyright International Business Machines Corporation 2015. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
IBM Redbooks promotions	vii
Preface	ix
Authors	ix
Now you can become a published author, too!	x
Comments welcome	x
Stay connected to IBM Redbooks	xi
Chapter 1. z/OS Management Facility	1
1.1 z/OSMF overview	2
1.1.1 z/OSMF summary of changes	3
1.2 z/OSMF initial configuration	5
1.2.1 Software prerequisites	5
1.2.2 z/OSMF setup	5
1.2.3 Adding plug-ins to your configuration	10
1.3 Upgrading z/OSMF from V1R13 and V2R1	14
1.3.1 Migrating from V1R13 to V2R2	14
1.3.2 Migrating from V2R1 to V2R2	15
1.4 New functions in z/OSMF V2R2	16
1.4.1 Multiple sysplex support	16
1.4.2 Incident Log aggregation	19
1.4.3 Configuring communication between primary and secondary z/OSMF	19
1.4.4 Enabling single sign-on between z/OSMF instances	20
Chapter 2. z/OS Common Event Adapter enhancements	23
2.1 CEA overview	24
2.1.1 CEA Time Sharing Option address space control	24
2.1.2 Benefit and value	25
Chapter 3. IBM Health Checker	27
3.1 Health Checker overview	28
3.1.1 New or updated health checks	28
3.2 REXX support for persistent data	33
3.2.1 Persistent data	34
3.2.2 Reading persistent data in REXX	35
3.2.3 Migration checks	36
Chapter 4. IBM predictive failure analysis	37
4.1 PFA overview	38
4.2 PFA summary of changes	38
4.2.1 Private storage exhaustion check	38
Chapter 5. Runtime diagnostics	43
5.1 RTD overview	44
5.2 Health-based routing integration with runtime diagnostics	46
Chapter 6. Subsystem initialization and management	47

6.1 Subsystem overview	48
6.2 Subsystem initialization problems	49
6.2.1 INITRTN pre-processing	49
6.2.2 SETSSI DELETE	50
6.2.3 EVENTRTN exit routine	52
6.2.4 DISPLAY SSI command	53
6.3 Benefit and value	53
Related publications	55
IBM Redbooks	55
Other publications	55
Help from IBM	55

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®	MVS™	WebSphere®
DB2®	Parallel Sysplex®	z Systems™
GDPS®	RACF®	z/OS®
IBM®	Redbooks®	z13™
IBM z Systems™	Redbooks (logo)  ®	zEnterprise®
IBM z13™	RMF™	
IMS™	System z®	

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

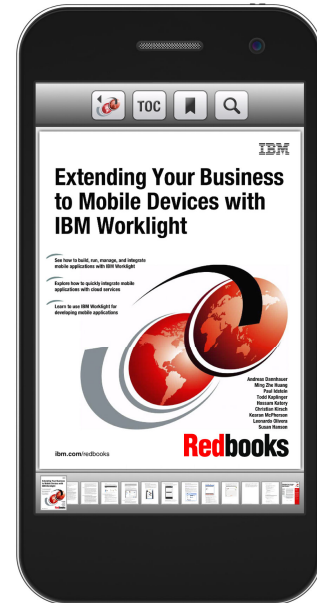
- ▶ Search, bookmark, save and organize favorites
- ▶ Get up-to-the-minute Redbooks news and announcements
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App



Download
Now

iOS



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!



ibm.com/Redbooks

About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication helps you to become familiar with the technical changes that were introduced into the Availability Management areas with IBM z/OS® V2R2.

This book is one of a series of IBM Redbooks publications that take a modular approach to providing information about the updates that are included with z/OS V2R2. This approach has the following goals:

- ▶ Provide modular content
- ▶ Group the technical changes into a topic
- ▶ Provide a more streamlined way of finding relevant information that is based on the topic

We hope you find this approach useful and we welcome your feedback.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Keith Winnard is the z/OS Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and is keen to engage with customers to understand what they want from IBM Redbooks Publications. Before joining the ITSO in 2014, Keith worked for clients and Business Partners in the UK and Europe in various technical and account management roles. He is experienced with blending and integrating new technologies into the traditional landscape of mainframes.

Wilson de Figueiredo is a z/OS System Programmer. He manages the operations support team at Banco do Brasil, a government bank in Brazil. He has more than 11 years of experience in mainframe systems. He holds a system analysis, internet consulting, and business administration degrees. His areas of expertise include IBM Parallel Sysplex®, z/OS security, and z/OS availability.

Redelf Janßen is a Client Technical Specialist in IBM z Systems™ Technical Sales Mainframe in Bremen, Germany. He holds a degree in Computer Science from the University of Bremen, and joined IBM in 1988. He is responsible for supporting z Systems clients in Germany. His areas of expertise include IBM z Systems hardware, z/OS, z/OSMF, storage management, and availability management. He has written extensively on IBM Redbooks since 1997. Since then, he co-authored many Redbooks publications on z/OS and both editions of z/OSMF.

Paulo Cesar Nascimento is a coordinator of support in the Eletrobras Eletronuclear plant in Brazil. He has 31 years of experience in mainframes. His areas of expertise include z/OS, IBM DB2®, SAP, DB, IBM RACF®, and z/OSMF. He has given talks at several IBM meetings, z/OS Explores, where customers share their successful projects.

Eweton Waki is an IT Analyst at Banco do Brasil, a government bank in Brazil. He has four years of experience in mainframe systems, including IBM GDPS®, DFSMS, high-end storage systems, and remote copy solutions. He holds a Bachelor's degree in Information Systems from Universidade Estadual de São Paulo. His area of expertise includes DFSMS and storage performance for mainframe environments.

Thanks to the following people for their contributions to this project:

Bob Haimowitz (Development Support Team [DST], Poughkeepsie Center) for setting up and maintaining the systems, and providing valuable advice, guidance, and assistance throughout the creation of this IBM Redbooks publication.

Rich Conway (DST, Poughkeepsie Center) for setting up and maintaining the systems, and providing valuable advice, guidance, and assistance throughout the creation of this IBM Redbooks publication.

Peter Bertolozzi (Systems Management specialist, IBM Redbooks residency support, Poughkeepsie Center) for setting up and maintaining the environments within syslab in which residents worked.

John Gierloff (Operations, Poughkeepsie Center) Residency set up and support.

Don Brennan (DST, Poughkeepsie Center) for setting up and maintaining the systems hardware used in the creation of this IBM Redbooks publication.

Ella Buslovich (Graphics specialist, location) for providing guidance and specialist graphics for this IBM Redbooks publication.

Ann Lund (ITSO Administration, Poughkeepsie Center) for administrative support to enable the residency's publication.

Cheryl Gera (ITSO Administration, Poughkeepsie Center) for managing the business operations for this IBM Redbooks publication.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



z/OS Management Facility

This chapter provides an introduction to the z/OS Management Facility (z/OSMF) and its new setup procedures for the use with z/OS V2R2. It also gives an overview of the new functions of z/OSMF.

This chapter includes the following topics:

- ▶ 1.1, “z/OSMF overview” on page 2
- ▶ 1.2, “z/OSMF initial configuration” on page 5
- ▶ 1.3, “Upgrading z/OSMF from V1R13 and V2R1” on page 14
- ▶ 1.4, “New functions in z/OSMF V2R2” on page 16

1.1 z/OSMF overview

IBM z/OS Management Facility (z/OSMF) is a browser-based system management tool for z/OS. It was originally introduced with z/OS V1R11 and then expanded with new functions over the last releases. z/OSMF helps you work with your z/OS environment in a simplified, optimized, and modern way. With z/OS V2R1, it was changed significantly because it was based on the IBM WebSphere® Application Server Liberty profile.

z/OSMF features new administrative functions in the following categories:

- ▶ Configuration, which includes the Configuration Assistant that is used for configuring the z/OS communication server
- ▶ Jobs and Resources, which features the option to include System Display and Search Facility (SDSF) via the z/OSMF Import Manager
- ▶ Links, in which you can configure web links for your own needs
- ▶ Performance, with which you can use for capacity provisioning, system status, resource monitoring, and workload management
- ▶ Problem Determination, which includes the Incident Log function
- ▶ Software, which includes the Software Management function
- ▶ z/OS Classic Interface, which includes the Interactive System Productivity Facility (ISPF) function
- ▶ z/OSMF Administration
- ▶ z/OSMF Settings

The z/OSMF Welcome page is shown in Figure 1-1.

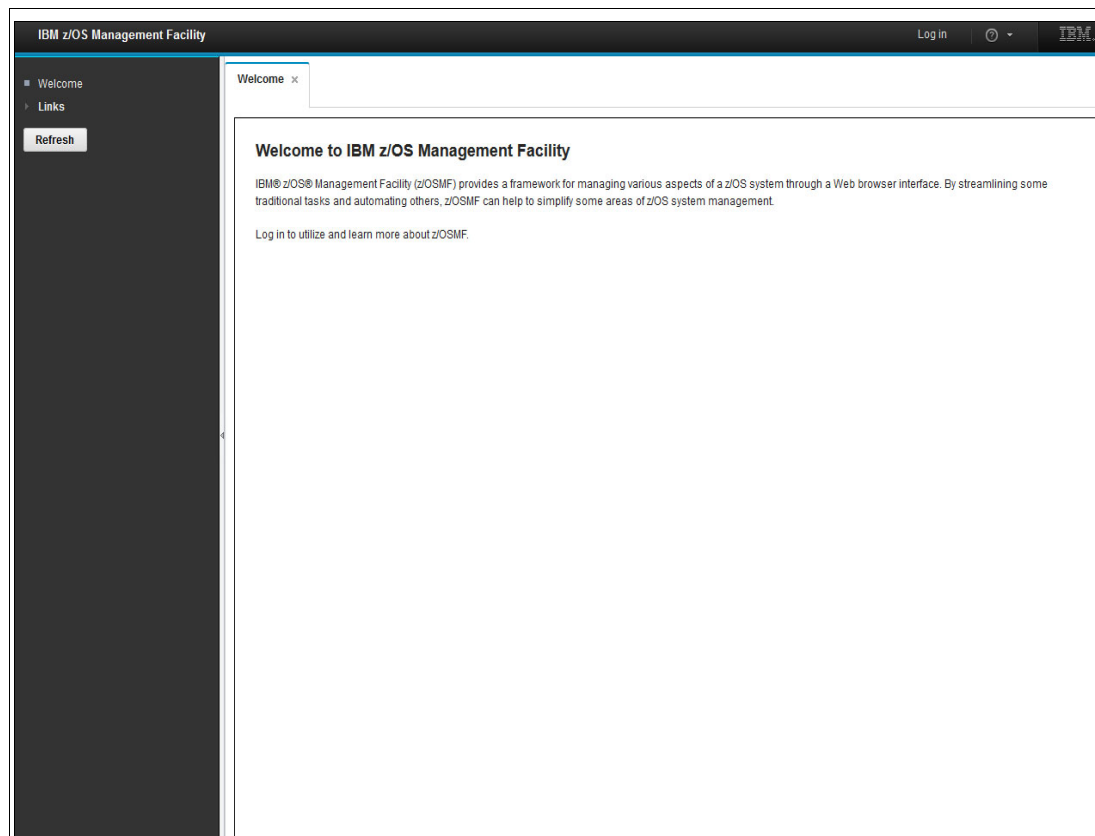


Figure 1-1 z/OSMF Welcome page after logging on

1.1.1 z/OSMF summary of changes

The following functions were introduced with IBM z/OS V2R2. The configuration process for z/OSMF changed starting with this release to provide an easier setup process:

- ▶ z/OSMF now is a base element of z/OS. Therefore, z/OSMF is not configured and ordered separately as it was in the past releases.
- ▶ z/OSMF now requires a new level of Java, which is the IBM 64-bit SDK for z/OS, Java Technology Edition, V7.1 (SR3), program number 5655-W44.
- ▶ If you prefer a simple configuration, you can use a simplified set of default values to set up your z/OSMF environment.
- ▶ If your configuration requires customization, this customization can be done by using a new IZUPRMxx parmlib member, which is used with the start procedure. You can find a sample member in SYS1.SAMPLIB(IZUPRM00). The content of this parmlib member is shown in Figure 1-2 on page 4. You can tailor this parmlib member for your own configuration.

```

HOSTNAME('*')
HTTP_SSL_PORT(2443)
INCIDENT_LOG UNIT('SYSALLDA')
JAVA_HOME('/usr/lpp/java/J7.1_64')
KEYRING_NAME('IZUKeyring.IZUDFLT')
LOGGING('*=warning:com.ibm.zosmf.*=info:com.ibm.zosmf.environment.ui=fi
ner')
RESTAPI_FILE ACCT(IZUACCT) REGION(32768) PROC(IZUFPROC)
SAF_PREFIX('IZUDFLT')
SEC_GROUPS USER(IZUUSER),ADMIN(IZUADMIN),SECADMIN(IZUSECAD)
SESSION_EXPIRE(495)
TEMP_DIR('/tmp')
UNAUTH_USER(IZUGUEST)
WLM_CLASSES DEFAULT(IZUGHTTP)
        LONG_WORK(IZUGWORK)

/* Uncomment the following statement and any plugins that
   are wanted */
/* PLUGINS( INCIDENT_LOG,
            COMMSERVER_CFG,
            WORKLOAD_MGMT
            RESOURCE_MON,
            CAPACITY_PROV,
            SOFTWARE_MGMT,
            ISPF)          */

```

Figure 1-2 Sample parmlib member for z/OSMF

- ▶ To simplify the configuration and future modifications for z/OSMF, the UNIX shell script `izusetup.sh` that you used in the past is now removed. For more information about the initial setup, see “z/OSMF initial configuration” on page 5. For more information about migration, see “Upgrading z/OSMF from V1R13 and V2R1” on page 14.
- ▶ If you are adding plug-ins after the initial setup, you can perform this task by using the z/OSMF Configuration Workflow. In previous releases, you performed this step by using the `izusetup.sh -add script`.
- ▶ In past releases, the security definitions were a result of the `izusetup.sh -config` step and your security administrator ran them as REXX scripts. This process is also modified with V2R2. Sample jobs are provided in `SYS1.SAMPLIB` and the related members feature the naming conventions `IZUxxSEC`.
- ▶ If you migrate from a previous z/OSMF release, you must use the migration process that was used in the previous releases. Run the `izumigrate.sh` script and the script now is enhanced to create a customized `IZUPRMxx` parmlib member for your environment. This parmlib member is based on the configuration settings from your earlier release. For more information about migration, see “Upgrading z/OSMF from V1R13 and V2R1” on page 14.

1.2 z/OSMF initial configuration

In this section, we describe the steps that you must perform when the base z/OSMF is initially set up in V2R2.

1.2.1 Software prerequisites

Because z/OSMF relies on Java, you first must ensure that IBM 64-bit SDK for z/OS, Java Technology Edition, V7.1 (SR3), program numBer 5655-W44, is installed and operational on your system. When you order z/OS V2R2, you also must order this release.

By default, the Java SDK is in its home directory `/usr/lpp/java/J7.1_64` on your system. If your installation has different naming conventions for the path, ensure that you adjust the path in the IZUPRMxx parmlib member. For more information, see Figure 1-2 on page 4.

For your user access to z/OSMF, we recommend the following browser releases:

- ▶ Microsoft Internet Explorer Version 9 or later
- ▶ Mozilla Firefox Version 17 or later

1.2.2 z/OSMF setup

In this section, we describe the steps to set up a z/OSMF configuration. Compared to the previous release, the setup steps are much easier to perform, which helps you to save time and to get your server running for the first time.

Setting up security

As a first step, set up the security definitions for the base, or task your security administrator to complete this process. You must set up these definitions once per sysplex or monoplex, depending on your environment. Copy the SYS1.SAMPLIB member IZUSEC to your JCL library, ensure that the default definitions fit your requirements and then, allow the job to run. Otherwise, ask your security administrator to adjust the definitions.

Allocating and mounting the z/OSMF file system

In the second step, the z/OSMF data file system is allocated and mounted. Copy SYS1.SAMPLIB member IZUMKFS to your own JCL library. Rename the ZFS data set if the predefined name does not fit your naming conventions.

The mount point for the data file system is `/var/zosmf`, which is the default that can be used in a single system environment. If you have a sysplex running where this zFS must be shared and automatically moved, update this job to ensure that you use a shared mount point.

When you run this job, ensure that your user ID includes superuser authority. This authority is necessary to mount the file system and issue the **MKDIR** and **CHMOD** commands inside the job.

Make note of the name of your data file system and its mount point and add this information to your BPXPRMxx parmlib member. This information ensures that your z/OSMF file system is mounted after an initial program load (IPL).

Starting the z/OSMF server

The third step is to start the z/OSMF server. There are two members in the IBM delivered SYS1.IBM.PROCLIB: IZUANG1 and IZUSVR1. Copy these members to your active proclib data set (the IZUSVR1 procedure is shown in Figure 1-5 on page 8).

Go to SYS1.SAMPLIB and copy the sample IZUPRM00 member that is shown in Figure 1-2 on page 4 to your parmlib. For a first run, we recommend that you verify that parameter HTTP_SSL_PORT(nnn) is defined with a valid TCP/IP port for z/OSMF. In our configuration, we use port 2443 to distinguish this port from the regular SSL port.

Modify the IZUSVR1 procedure, changing the parameter from IZUPRM='NONE' to IZUPRM='xx', where xx is your parmlib member suffix. Then, start the two tasks by using the following order:

```
S IZUANG1,JOBNAME=jobname
S IZUSVR1,JOBNAME=jobname
```

Alternatively, you also can start the procedures and omit the jobname specification. In this case, use the following procedures:

```
S IZUANG1
S IZUSVR1
```

The method that is used to start the procedures depends on your system and security-related settings.

The SYSLOG or OPERLOG entries that are generated during the server start are shown in Figure 1-3.

```
S IZUSVR1
IRR812I PROFILE IZUSVR1.* (G) IN THE STARTED CLASS WAS USED 372
      TO START IZUSVR1 WITH JOBNAME IZUSVR1.
$HASP100 IZUSVR1 ON STCINRDR
IEF695I START IZUSVR1 WITH JOBNAME IZUSVR1 IS ASSIGNED TO USER
IZUSVR , GROUP IZUADMIN
$HASP373 IZUSVR1 STARTED
+IEE252I MEMBER IZUPRM00 FOUND IN SYS1.PARMLIB
IZUG400I: The z/OSMF web application services are initialized.
+CWWKF0011I: The server zosmfServer is ready to run a smarter planet.
```

Figure 1-3 IZUSVR1 server start messages in SYSLOG or OPERLOG

Monitor the IZUSVR1 joblog and its output during server start for the message that is shown in Figure 1-4. Use the URL that is shown in Figure 1-4 to connect to your z/OSMF instance. The port that is shown is the port that we defined in IZUPRMxx parmlib member.

```
IZUG349I: The z/OSMF Server home page can be accessed at
      : https://WTSC76.ITS0.IBM.COM:2443/zosmf
      : after the z/OSMF server is started on your system.
```

Figure 1-4 IZUG349I message for IZUSVR1 server in JOBLOG

As shown in Figure 1-5 on page 8, the z/OSMF V2R2 IZUSVR1 start procedure is different from the procedure that used in z/OSMF V2R1. The new procedure includes the following steps:

- ▶ The ZPARM step that parses the new IZUPRMxx parmlib member. This step is new because a parmlib member is used.
- ▶ The CONFZMF step that configures the z/OSMF server. This step also is new and completes some of the work that in the previous release was done by using the `izusetup.sh` shell script. However, this configuration setup is done whenever you start the server.
- ▶ The ZOSMF step that starts the WebSphere Liberty Profile server. In the WLPUDIR DD statement, it points to the `/var/zosmf/configuration` user directory that is specified in the USERDIR variable on the PROC statement.

The IZUSVR1 procedure now also supports the following parameters that you can check before starting the server and change, if needed:

- ▶ **ROOT**
This parameter points to the z/OSMF code directory path `/usr/lpp/zosmf` by default. It must be enclosed in quotation marks, start with a slash (`/`), and be fully qualified. The version and release are no longer part of its name.
- ▶ **OUTCLS**
By using this parameter, you can select an output class for writing z/OSMF system output. By default, it is class `*`. You must specify the output class in quotation marks.
- ▶ **USERDIR**
This parameter is the z/OSMF user directory path. By default, it is `/var/zosmf`. This path is new and it is no longer `/etc/zosmf`, as it was in previous releases. The parameter must be enclosed in quotation marks, start with a slash (`/`), and be fully qualified.
- ▶ **TRACE**
You can use this parameter to enable tracing for configuration-related issues. The error data is written to the server job log. You activate tracing only at the request of IBM support. By default, tracing is set to `"N"` and can be enabled by changing the value to `"Y"`.
- ▶ **IZUPRM**
This parameter manages the parameterization of the z/OSMF instance and brings that instance closer to z/OS. It replaces the `izuconfig` file that you used in past releases of z/OSMF.

By default, this parameter is not used and it has the value `IZUPRM='NONE'`. If you use IZUPRMxx members, specify this parameter by using the form `IZUPRM='xx'` or `IZUPRM=(xx,yy,zz)`. If you specify one or more suffixes, these members must be in your system's parmlib concatenation. The syntax of the IZUPRMxx member is shown in Figure 1-2 on page 4.
- ▶ **IZUMEM**
By using this parameter, you can define the maximum amount of virtual storage for the z/OSMF server address space that is above the bar. It is used for the IZUSVR1 procedure only. By default, the limit is 4 GB. You can specify `IZUMEM='NOLIMIT'`.

```

//IZUSVR1  PROC PARMS='zosmfServer',      /* Server parms */
//          ROOT='/usr/lpp/zosmf',        /* zOSMF install root */
//          OUTCLS='*',                  /* Sysout class */
//          USERDIR='/var/zosmf',        /* Config dir */
//          TRACE='N',                  /* Trace option */
//          IZUPRM='00',                /* Parmlib suffixes or NONE */
//          IZUMEM=4G                   /* Server memlimit */
//
//
//-----
/* Parse z/OSMF PARMLIB member
//-----
//ZPARM  EXEC  PGM=IZUPARMS,REGION=OM,
// PARM='/IZUPRM=&IZUPRM,TRACE=&TRACE,USERDIR=&USERDIR'
//DFLTCFG DD  PATH='&ROOT./defaults/configuration.defaults'
//STDOUT  DD  SYSOUT=&OUTCLS
//STDERR  DD  SYSOUT=&OUTCLS
//CEEDUMP DD  SYSOUT=&OUTCLS
//
//-----
/* Configure z/OSMF server
//-----
//CONFZMF EXEC PGM=BPXBATCH,REGION=OM,COND=(0,LT),
// PARM='SH &ROOT./bin/izuconfig.sh &ROOT &USERDIR &TRACE'
//SYSPRINT DD SYSOUT=&OUTCLS
//SYSOUT   DD SYSOUT=&OUTCLS
//STDERR   DD SYSOUT=&OUTCLS
//STDOUT   DD SYSOUT=&OUTCLS
//
//-----
/* Start the Websphere Liberty Profile server
//
/* WLPUDIR - PATH DD that points to the Liberty Profile's "user"
/*          directory. If the DD is not allocated, the user
/*          directory location defaults to the wlp/usr directory
/*          in the install tree.
/* STDOUT - Destination for stdout (System.out)
/* STDERR - Destination for stderr (System.err)
/* STDENV - Initial UNIX environment - read by the system. The
/*          installation default and server specific server
/*          environment files will be merged into this environment
/*          before the JVM is started.
//
//-----
//ZOSMF  EXEC PGM=BPXBATSL,REGION=OM,COND=(0,LT),
// MEMLIMIT=&IZUMEM.,TIME=NOLIMIT,
// PARM='PGM &ROOT./wlp/lib/native/zos/s390x/bbgzsrv --clean &PARMS'
//
//WLPUDIR DD PATH='&USERDIR./configuration'
//
//STDOUT  DD SYSOUT=&OUTCLS
//STDERR  DD SYSOUT=&OUTCLS
//*STDOUT DD PATH='&ROOT/izusvr1.stdout',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU

```

Figure 1-5 IZUSVR1 start procedure with active IZUPRM00 parmlib member

If you want to stop the server later, enter the following commands:

```
P IZUSVR1  
P IZUANG1
```

You must enter the **STOP** commands in this sequence to avoid hang situations. If the **STOP** commands fail, enter the **CANCEL** commands.

Logging on to your z/OSMF instance

Now that a base configuration is complete, you can use it to log on. Use one of the supported browsers, open a window, and enter the URL that was shown in your JOBLOG. We use the following URL in our implementation:

`https://wtsc76.itso.ibm.com:2443/zosmf`

Click **Log in** and enter your user ID and password, as shown in Figure 1-6.

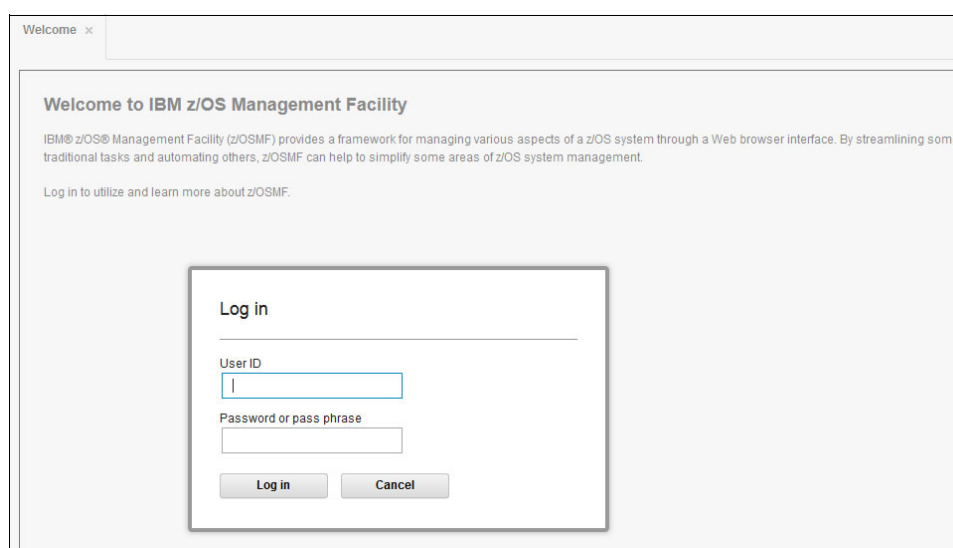


Figure 1-6 z/OSMF Log In window

Another new function on the main panel is that you can change your password or switch to another user without in your z/OSMF session, as shown in Figure 1-7.

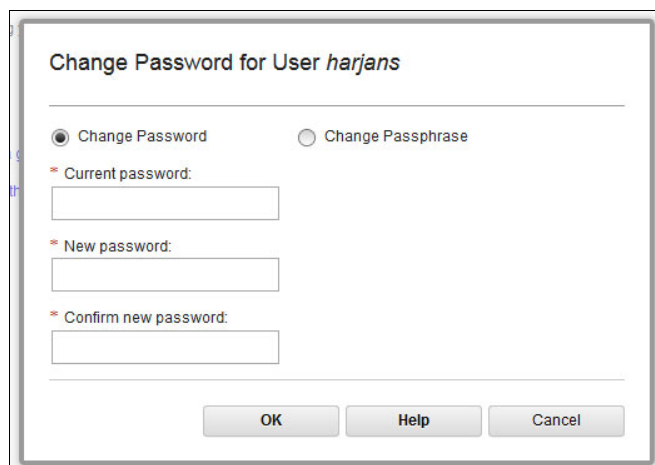


Figure 1-7 Change Password for User window

This change goes directly to your RACF database, which means that your next TSO logon must include the updated password.

1.2.3 Adding plug-ins to your configuration

Your next step in configuring your z/OSMF instance is to add the plug-ins that you need for you and your colleagues' daily work. In this section, we describe the steps to perform.

When you use plug-ins, the plug-ins are configured by using z/OSMF Workflow. This configuration is a major change compared to the previous release.

First, you must ensure that your user ID has access to various profiles from the FACILITY class. When you ran the IZUSEC job as described in "Setting up security" on page 5, you updated the definitions, as shown in Figure 1-8.

```
/* Allow users of the z/OSMF Configuration Workflow to extract      */
/* profile information                                              */
RDEFINE FACILITY IRR.RADMIN.LISTUSER
RDEFINE FACILITY IRR.RADMIN.LISTGRP
RDEFINE FACILITY IRR.RADMIN.RLIST
RDEFINE FACILITY IRR.RADMIN.SETROPTS.LIST

/* Permit the z/OSMF administrator access                        */
PERMIT IRR.RADMIN.LISTUSER CLASS(FACILITY) ID(IZUADMIN) +
  ACCESS(READ)
PERMIT IRR.RADMIN.LISTGRP CLASS(FACILITY) ID(IZUADMIN) +
  ACCESS(READ)
PERMIT IRR.RADMIN.RLIST CLASS(FACILITY) ID(IZUADMIN) +
  ACCESS(READ)
PERMIT IRR.RADMIN.SETROPTS.LIST CLASS(FACILITY) ID(IZUADMIN) +
  ACCESS(READ)
```

Figure 1-8 RACF commands for authorizing the users of Configuration Workflow

You then must import the XML-based workflow definition file into the Workflows task. The XML file is available in the following directory:

```
/usr/lpp/zosmf/workflow/izu.config.setup.xml
```

In addition to the XML definition file, you must specify the corresponding variable input file, which by default is in the following directory:

```
/var/zosmf/configuration/workflow/izu.config.workflow.cfg
```

This file contains variables that include path names and configuration file names. You see its default that was created during the base setup, as shown in Figure 1-9.

```
IZU_CODE_ROOT=/usr/lpp/zosmf
IZU_LOGFILE_DIR=/var/zosmf/data/logs
IZU_CONFIG_DIR=/var/zosmf/configuration
IZU_CONFIG_FILE=/var/zosmf/configuration/active_configuration.cfg
JAVA_HOME=/usr/lpp/java/J7.1_64
PEGASUS_HOME=/usr/lpp/wbem
IZU_CONFIG_FILE_BASENAME=active_configuration.cfg
```

Figure 1-9 Variable input file for plug-in setup workflow

With this information, you can start preparing your configuration workflow. Complete the following steps:

1. Click **Workflows** in your z/OSMF browser workspace. A tab opens.
2. Click **Action** → **Create Workflow**.

The Create Workflow window opens that includes two steps in which you enter the paths of the XML-based definition file and variable input file, as shown in Figure 1-10.

Workflows
Simplifies tasks through guided step-based workflows, and provides administrative functions for assigning workflow responsibilities and tracking progress.

Actions ▾

✚ ➡ No filter applied

Workflow Name Filter

Create Workflow

Type or select a workflow definition file to use for creating a new workflow.
For a z/OS data set, specify a fully qualified name, with no quotes.

Workflow definition file:
/usr/lpp/zosmf/workflow/izu.config.setup.xml

Type or select a variable input file to populate the new workflow. For a z/OS data set, specify a fully qualified name, with no quotes.

Workflow variable input file:
/var/zosmf/configuration/workflow/izu.config.workflow.cfg

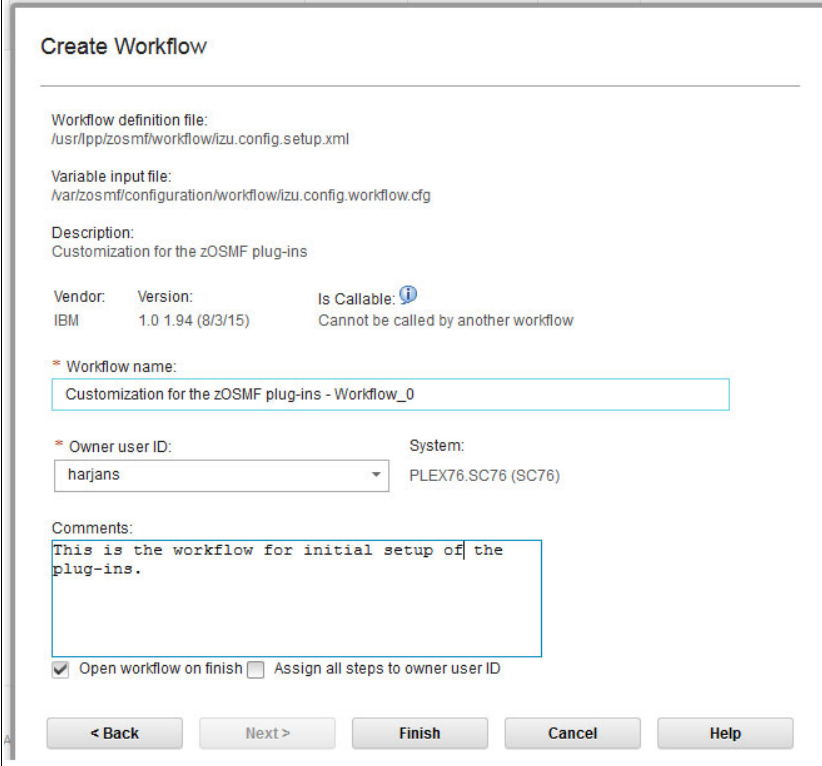
System:
PLEX76.SC76 (SC76)

< Back Next > Finish Cancel Help

Figure 1-10 Create Workflow window

3. After the information is entered in the first window, click **Next**.

4. In the next window, enter a system name, workflow name, owner user ID, and optionally, a comment, as shown in Figure 1-11.




The 'Create Workflow' window displays pre-filled information for a workflow named 'Customization for the zOSMF plug-ins'. It includes fields for workflow definition file, variable input file, description, vendor (IBM), version (1.0 1.94), and a note that it cannot be called by another workflow. Required fields for 'Workflow name' and 'Owner user ID' (set to 'harjans') are present, along with a 'System' field set to 'PLEX76.SC76 (SC76)'. A 'Comments' text area contains the text 'This is the workflow for initial setup of the plug-ins.' At the bottom, there are checkboxes for 'Open workflow on finish' (checked) and 'Assign all steps to owner user ID' (unchecked), followed by navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Create Workflow

Workflow definition file:
/usr/lpp/zosmf/workflow/izu.config.setup.xml

Variable input file:
/var/zosmf/configuration/workflow/izu.config.workflow.cfg

Description:
Customization for the zOSMF plug-ins

Vendor: IBM Version: 1.0 1.94 (8/3/15) Is Callable:  Cannot be called by another workflow

* Workflow name:
Customization for the zOSMF plug-ins - Workflow_0

* Owner user ID: harjans System: PLEX76.SC76 (SC76)

Comments:
This is the workflow for initial setup of the plug-ins.

☒ Open workflow on finish ☐ Assign all steps to owner user ID

< Back Next > Finish Cancel Help

Figure 1-11 Create Workflow window

- After all of the information is entered, click **Finish**. The workflow is created as shown in Figure 1-12. You can now start setting up the various plug-ins.

Customization for the zOSMF plug-ins - Workflow_0

Description: Customization for the zOSMF plug-ins
Percent complete: 0%

Owner: harjans
Steps complete: 0 of 98

System: PLEX76.SC76 (SC76)
Status: In Progress

Is Callable: Cannot be called by another workflow

Workflow Steps

State Filter	No. Filter	Title Filter	CalledWorkflow Filter	Automated Filter	Owner Filter	Skill Category Filter
<input type="checkbox"/>	3	Choose the optional plug-ins to be added		No		System Programmer
<input type="checkbox"/>	4	Ensure that CEA common event adapter (CEA) is active				
<input type="checkbox"/>	5	Common Information Model (CIM) server				
<input type="checkbox"/>	6	Configuration Assistant plug-in				
<input type="checkbox"/>	7	ISPF plug-in				
<input type="checkbox"/>	8	Workload Management plug-in				
<input type="checkbox"/>	9	Resource Monitoring plug-in				
<input type="checkbox"/>	10	Capacity Provisioning plug-in				
<input type="checkbox"/>	11	Software Deployment plug-in				
<input type="checkbox"/>	12	Incident Log plug-in				
<input checked="" type="checkbox"/>	13	Re-start the z/OSMF server		No		System Programmer

Total: 125 Selected: 1

[Return to Workflows](#) [Refresh](#) Last refresh: Aug 25, 2015, 4:11:44 PM local time (Aug 25, 2015, 8:11:44 PM GMT)

Figure 1-12 Main panel for the plug-in customization workflow

Depending on the number of plug-ins you want to install, repeat these steps and always check the results that are shown after you run the Perform step. There are several **DISPLAY** commands that are performed within the steps. You must check their output and perform corrections, if necessary.

Note: Verify with your security expert that you have the RACF class SDSF activated. Also, verify that the user ID that performs the workflow steps has READ access to profile ISFOPER.SYSTEM. Otherwise, commands that are submitted by specific examination steps of the workflow fail. Use the following commands to grant access to the operator commands:

```
RDEFINE SDSF ISFOPER.SYSTEM UACC(NONE)
PERMIT ISFOPER.SYSTEM CLASS(SDSF) ID(IZUADMIN) ACCESS(READ)
SETROPTS RACLIST(SDSF) REFRESH
```

Depending on your z/OS environment, you might not yet activate RACF class SDSF because of the use of SDSF ISFPRMxx parmlib member.

After you finish configuring the plug-ins and their prerequisites, complete the following steps to finish the process:

- Modify your IZUPRMxx parmlib member. Uncomment the PLUGIN statement and its values for the plug-ins that you configured during the workflow.
- Restart your z/OSMF server to activate the new plug-ins.

1.3 Upgrading z/OSMF from V1R13 and V2R1

Because z/OS V2R2 can run with V1R13 and V2R1, you can perform a z/OSMF migration from both releases. We describe the migrations process in the next sections and describe the steps for migrating from V2R1 to V2R2. We recommend that you follow the migration steps that are described in *Migration from z/OS V2R1 and z/OS V1R13 to z/OS V2R2*, GA32-0889.

1.3.1 Migrating from V1R13 to V2R2

When your source system is running z/OS V1R13, you must complete the following migration tasks:

1. Convert to SAF Authorization Mode. If your system still runs the old Repository Authorization Mode, you must first convert your security setup to SAF Authorization Mode in V1R13. The z/OSMF configuration process generates the relevant REXX procedures that your security administrator must run.
2. Review the SAF profile prefix. The configuration variable IZU_WAS_PROFILE_PREFIX with its value BBNBASE was used until V1R13. Since V2R1, it is no longer used. Instead, a new configuration variable IZU_SAF_PROFILE_PREFIX is in place since V2R1. By default, its value is IZUDFLT. You can later remove the BBNBASE profiles. Use the following RACF command to find all affected profiles in your RACF database:

```
SEARCH ALL CLASS(ZMFAPLA) FILTER(BBNBASE.**)
```
3. Check the security for ports 32207 and 32208. With V2R2, these old ports are no longer used because now z/OSMF by default uses port 443 for SSL communication. If there are conflicts, you can define another port. You might want to contact your communication server colleague first.
4. Evaluate the usage of user ID ZOSMFAD. This user ID is no longer used. If you do not need to keep this ID for other purposes, we recommend that you to remove it.
5. Remove WebSphere constructs from previous releases. Starting with z/OSMF V2R1, there was no need for the IBM WebSphere Application Server OEM Edition for z/OS because the base changed to IBM WebSphere Liberty profile. Therefore, you can remove old constructs that are in your file systems. Check files and directories under the /zWebSphereOEM/V7R0/config1 directory. You might also check your SYS1.PROCLIB data set for members starting with BBN7*. You can also remove them after you migrate to V2R2.
6. Remove the APF authorization for SYS1.MIGLIB(AMATERSE). Starting with z/OSMF V2R1, it is no longer necessary to keep SYS1.MIGLIB APF-authorized for Incident Log. Therefore, if there is no other exploiter, you can remove this library from APF authorization.
7. Remove the most-generic profile for z/OSMF authorizations. With z/OSMF V2R2, profiles in RACF class ZMFAPLA are no longer created as generic profiles as it was done previously. The job in SYS1.SAMPLIB(IZUSEC) now provides authorizations that are more granular and based on discrete profiles. Use the **SEARCH ALL CLASS(ZMFAPLA) FILTER(IZUDFLT.**)** RACF command to find all affected profiles in your RACF database.

Use the following RACF commands to remove the most-generic profile:

```
RDELETE ZMFAPLA IZUDFLT.ZOSMF.**  
SETROPTS RACLIST(ZMFAPLA) REFRESH
```

Although the generic profiles still work, the new, more discrete profile provides you a more granular access to specific resources in z/OSMF.

8. Check the security profile for the Software Management task. Because of name changes in z/OSMF V2R1, you must use create a generic RACF profile, such as `IZUDFLT.ZOSMF.SOFTWARE_DEPLOYMENT.**`, or a discrete profile, such as `IZUDFLT.ZOSMF.SOFTWARE_DEPLOYMENT.SOFTWARE_MANAGEMENT` to control access to the Software Management task. Use the following RACF command to determine your settings:

```
SEARCH ALL CLASS(ZMFAPLA) FILTER(IZUDFLT.**)
```

9. Authorize the z/OSMF server to create PassTickets. If you use the Capacity Provisioning plug-in or the Resource Monitoring plug-in in z/OSMF and these functions use PassTickets, you must change the server user ID in the RACF profiles from the old default `WSSRU1` to the new default `IZUSVR`, or your equivalent name. Use the following RACF commands to grant access:

```
PERMIT IRRPTAUTH.CFZAPPL.* CLASS(PTKTDATA) ID(IZUSVR) ACCESS(UPDATE)
SETROPTS RACLIST(PTKTDATA) REFRESH
PERMIT IRRPTAUTH.GPMSEVR.* CLASS(PTKTDATA) ID(IZUSVR) ACCESS(UPDATE)
SETROPTS RACLIST(PTKTDATA) REFRESH
```

10. Install the z/OSMF cataloged procedures. You find the procedures `IZUANG1` and `IZUSVR1` in `SYS1.IBM.PROCLIB`. Copy the members to your proclib for z/OS V2R2. Also, copy `IZUFPROC` from `SYS1.IBM.PROCLIB` to your preferred proclib. You need this procedure for the z/OS data set and REST interface in z/OSMF.

11. Verify that the z/OSMF server has sufficient authorization.

12. Define the z/OSMF started procedure to RACF.

13. Migrate to the new release of z/OSMF. You migrate to z/OSMF V2R2 by using the script `izumigrate.sh`, which you might use in previous migrations. In z/OS V2R2, this script is enhanced to create the `IZUPRMxx` parmlib member that is shown in Figure 1-2 on page 4. For more information about the migration process, see “Migrating from V2R1 to V2R2” on page 15.

14. Review the new z/OSMF service process. In previous releases, you sometimes had to run the `izusetup.sh` script with the `-service` option to apply service to your z/OSMF server. z/OSMF V2R1 changed this behavior so that a server restart satisfied the service process. The appropriate APAR cover letters have detailed descriptions of the actions.

For more information about migration, see *Migration from z/OS V2R1 and z/OS V1R13 to z/OS V2R2*, GA32-0889.

1.3.2 Migrating from V2R1 to V2R2

When your base system was on z/OS V2R1, you were at a good starting point. You can then perform the migration from z/OSMF after the initial load of your z/OS V2R2 system.

We describe the necessary steps for this migration in this section.

Note: Your z/OSMF server must be down during migration.

Use the script `izumigrate.sh` to migrate different configuration values from your previous configuration file (which is by default `izuconfig1.cfg`) to the parmlib member `IZUPRMxx`.

Customize and run the sample jobs IZUSEC and IZUxxSEC that are provided in your V2R2 SYS1.SAMPLIB to get the necessary security definitions for your z/OSMF V2R2 environment. There might be some definitions that are active in your z/OSMF V2R1 environment. When this situation occurs, your joblog shows several RACF ICH408I messages.

The next step is to create a copy of SYS1.SAMPLIB(IZUMKFS). As described in “Allocating and mounting the z/OSMF file system” on page 5, this step is necessary to allocate, format, and mount the new data file system for z/OSMF.

In z/OSMF V2R2, the name of the data file system is IZU.SIZUUSRD. Ensure that this name fits your naming conventions; otherwise, rename it to an appropriate data set name. There is another step with name MIGRATE that you use for your migration. This step copies the content of your old file system to the new one.

If your old file system is already mounted at /var/zosmf/data, unmount it and then, mount it on a different mount point because the new file system now uses /var/zosmf/data in V2R2. Modify IZUMKFS to your naming conventions and run the job.

The names of the start procedures are the same as in V2R1. IZUANG1 is the procedure to the angel process and IZUSVR1 is the procedure to start the server instance. Copy both procedures from SYS1.IBM.PROCLIB to your own PROCLIB.

Important: Do not use your old V2R1 procedures because doing so leads to failures during z/OSMF start. There is a new parameterization in the V2R2 procedures.

On your z/OS console, use the following procedures to start your z/OSMF instance:

```
S IZUANG1  
S IZUSVR1
```

or

```
S IZUSVR1,IZUPRM=xx
```

1.4 New functions in z/OSMF V2R2

In addition to the new method for configuration, z/OSMF offers you some other functions that are described in this section.

1.4.1 Multiple sysplex support

Until V2R1, z/OSMF was a management application with single sysplex scope. Software management was one function that worked across sysplex boundaries.

z/OSMF V2R2 now provides a common framework that supports the management of multiple sysplex environments from a single z/OSMF browser instance. Therefore, each sysplex must run its own z/OSMF instance. The z/OSMF instance that you connect to is then called as the primary z/OSMF. You can then manage other z/OSMF instances, which are called remote z/OSMF or secondary z/OSMF.

You might know these terms from remote software deployment. An example of a multi-sysplex z/OSMF environment is shown in Figure 1-13.

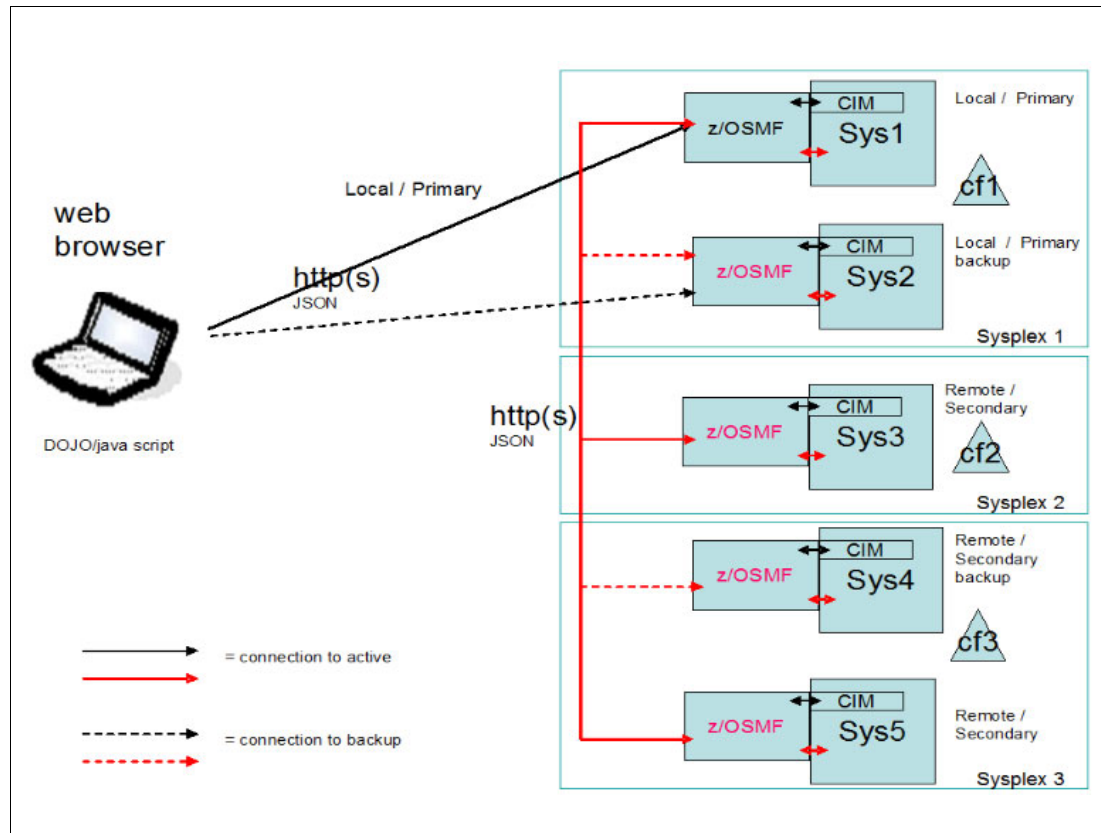


Figure 1-13 Layout of a z/OSMF multi-sysplex environment

With this release, a common framework was introduced that supports the management of multiple sysplex, including the following support:

- ▶ The z/OSMF systems task now manages the topology information of z/OS systems, sysplex, Central Processor Complex (CPC), and groups of systems.
- ▶ A graphical view of the topology and the ability to export the graphical view.
- ▶ A topology REST interface with which you can work to get topology information, such as groups and sysplex through the invocation of the z/OSMF RESTful API.
- ▶ A z/OSMF navigation tree that provides the option to open a plug-in with the following scopes:
 - Single sysplex
 - CPC
 - Specific group
- ▶ A multi-system routing REST service with which you can communicate with single remote systems or groups of remote systems.
- ▶ The capability for Single Sign On (SSO) that avoids the need for logging in to each remote or secondary z/OSMF instance when you manage it from your primary z/OSMF instance.

For this function, your primary z/OSMF instance must run on V2R2. All of these functions (except the support of a graphical topology view and the navigation tree) also can run on a z/OSMF V2R1 remote or secondary system.

In the z/OSMF Systems task, you can now view the topology in different ways. The traditional view gives you the ability to switch between sysplex, group, or CPC. The sysplex-based view is shown in Figure 1-14.

Systems

View systems by: Sysplex

Actions | Table view: Tree

No filter applied

Sysplex/System Name	Nickname	Description	Groups	URL	z/OS Version and Release	JES Member Name	JES Type	CPC Name
WTSCPLX8								
SC81	SC81	SC80/81		https://wtsc81.tso.ibm.com:2443/zoamf	z/OS V2R2	SC81	JES2	SCZP501
SC80	SC80				z/OS V2R1	SC80	JES2	SCZP501
PLEX76								

Figure 1-14 z/OSMF topology by sysplex

With the graphic view support, you can now review the topology in a new way. The criteria is also sysplex, group, or CPC. The topology by CPC is shown in Figure 1-15.

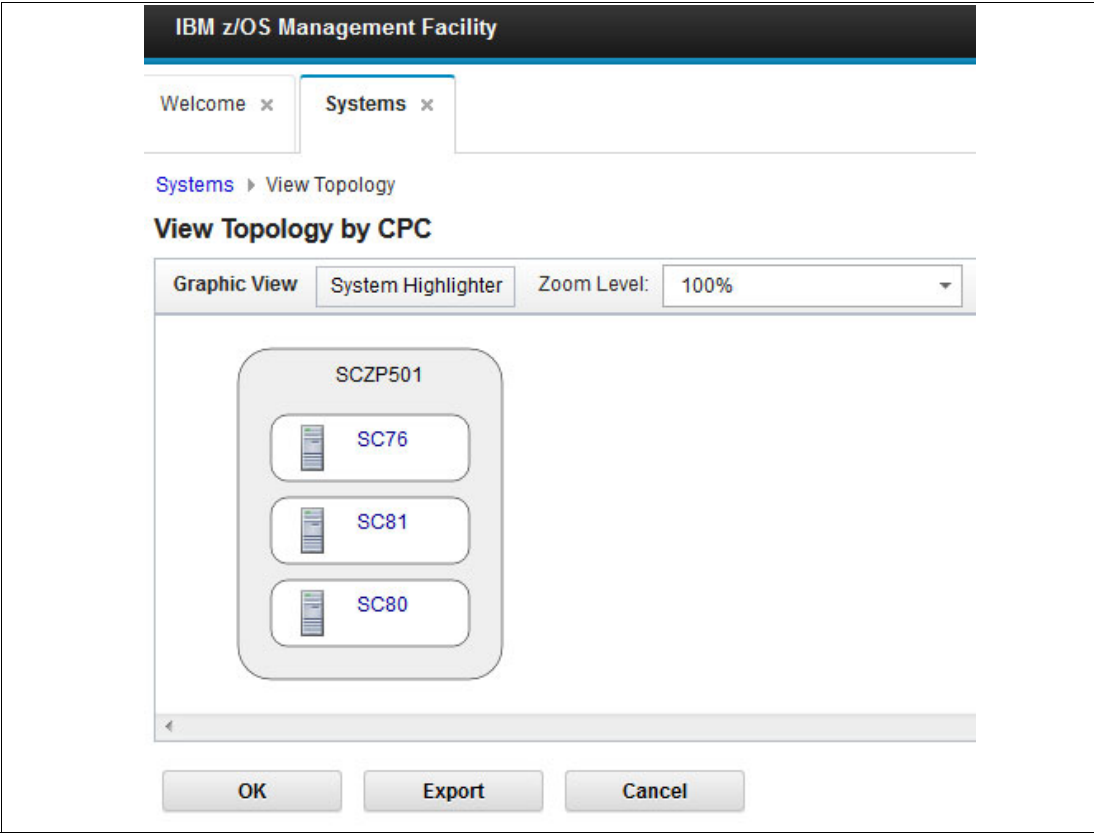


Figure 1-15 z/OSMF graphical representation of a topology by CPC

You can export this view in an XML-based scalable vector graphics format (svg) for later use in other software, such as a browser or graphics program.

Enablement for SSO

The SSO technique enables you to log on to your primary z/OSMF instance and get access to other (secondary) z/OSMF instances, as shown in Figure 1-13 on page 17. z/OSMF uses the Lightweight Third Party Authentication (LTPA) security protocol to enable a secure single-sign on environment. For more information about this setup, see “Enabling single sign-on between z/OSMF instances” on page 20.

1.4.2 Incident Log aggregation

Incident Log task was introduced as one of the first plug-ins in z/OSMF V1R11. Since then, various enhancements were made, most based on client requirements. The Incident Log plug-in helps you to manage your z/OS-based incidents and those incidents coming from other subsystems, such as IBM IMS™, or DB2. If it is enabled through the configuration workflow, it automatically captures memory dumps and diagnostic data and allows you to send this data to the support center.

Before z/OSMF V2R2, Incident Log was single sysplex scope, meaning that you managed each of your sysplex environments separately. By using the Multiple Sysplex support in z/OSMF V2R2, Incident Log allows you to manage incidents from a single or multiple sysplex environments in one browser window. This session is connected to your primary z/OSMF instance only. The Incident Log plug-in now supports the following modes:

- ▶ Basic-Proxy mode

Incident Log is used with scope of a single sysplex or single system.

- ▶ Aggregation mode

Incident Log is used in a group that contains multiple sysplex or multiple systems.

If you plan to use this cross z/OSMF function, you must establish the communication between your primary and secondary z/OSMF instances. For more information, see 1.4.3, “Configuring communication between primary and secondary z/OSMF” on page 19.

1.4.3 Configuring communication between primary and secondary z/OSMF

You can configure your z/OSMF to communicate with another instance of z/OSMF in a remote sysplex, monoplex, or local system. This capability can be important because tasks, such as Incident Log, can be used to work with remote systems by using a single-sign on. We describe such a setup in this section.

To enable communication between different z/OSMF systems, you must configure a primary z/OSMF instance for communication with other, secondary instances. The key requirement is to enable the sharing of digital certificates between the participating instances.

During the configuration process of a z/OSMF server instance, the security job IZUSEC creates a certificate authority (CA) and a server certificate. These certificates are used for enabling SSL connections between the z/OSMF instances. In this job, z/OSMF also creates a SAF key ring and stores the CA and server certificate in the key ring. Figure 1-16 on page 20 shows the RACF definition for the CA certificate.

```

/* Create the CA certificate for the z/OSMF server
RACDCERT CERTAUTH GENCERT +
  SUBJECTSDN(CN('z/OSMF CertAuth for Security Domain') +
    OU('IZUDFLT')) WITHLABEL('zOSMFCA') +
  TRUST NOTAFTER(DATE(2023/05/17))
RACDCERT ADDRING(IZUKeyring.IZUDFLT) ID(IZUSVR)

```

Figure 1-16 RACF definition for creating the CA certificate

Figure 1-17 shows the RACF definition for the server certificate.

```

/* Create the server certificate for the z/OSMF server */
/* Change HOST NAME in CN field into real local host name */
/* Usually the format of the host name is 'XXXX.XXX.XXX.XXX' */
RACDCERT ID( IZUSVR ) GENCERT SUBJECTSDN(CN('WTSC76.ITSO.IBM.COM') +
  O('IBM') OU('IZUDFLT')) WITHLABEL('DefaultzOSMFCert.IZUDFLT'), +
  SIGNWITH(CERTAUTH LABEL('zOSMFCA')) NOTAFTER(DATE(2023/05/17))
RACDCERT ALTER(LABEL('DefaultzOSMFCert.IZUDFLT')) ID(IZUSVR) TRUST
RACDCERT ID( IZUSVR ) CONNECT (LABEL('DefaultzOSMFCert.IZUDFLT') +
  RING(IZUKeyring.IZUDFLT) DEFAULT)
RACDCERT ID( IZUSVR ) CONNECT (LABEL('zOSMFCA') +
  RING(IZUKeyring.IZUDFLT) CERTAUTH)

```

Figure 1-17 RACF definition for creating the server certificate

After submitting these definitions, run the RACF **SETROPTS RACLIST(FACILITY) REFRESH** command to activate them.

When you plan for a secure communication between z/OSMF instances, you have the following options for sharing CA certificates because CA certificates are used to sign the server certificates:

- ▶ SSL connections that use the same CA certificate. You might use this option if you start implementing z/OSMF and no secondary instances exist. In this case, the server certificates in the primary and secondary instances are created by using the same CA.
- ▶ SSL connections by using different CA certificates. You might use this option if secondary instances are available. In this case, each instance uses its own CA and CA certificate to sign its own server certificate. You then perform the following tasks:
 - Export the CA certificate from each secondary system.
 - Import the CA certificates into the primary system security database.
 - Connect the CA certificates to the primary system.

1.4.4 Enabling single sign-on between z/OSMF instances

In preparation for multi-sysplex support, z/OSMF V2R1 with SPE3 introduced the Lightweight Third Party Authentication (LTPA) security protocol, which is provided by WebSphere Application Server Liberty profile to enable a secure single sign-on environment among multiple z/OSMF instances. It enables you to log on to one z/OSMF instance and access other z/OSMF instances without seeing more logon prompts.

The LTPA protocol uses an LTPA token to authenticate a user with the z/OSMF servers that are enabled for single sign-on. The token contains information about the user and is encrypted by using a cryptographic key. The participating z/OSMF servers pass the token to other z/OSMF servers through cookies for web resources. If the receiving server uses the same key as the primary z/OSMF server, it decrypts the token to obtain the user information, verifies that the token is not expired, and confirms that the user ID is in its user registry.

The primary z/OSMF server is the server that generated the key to be used for single sign-on. After the receiving server validates the LTPA token, the server authenticates the user with that z/OSMF instance and allows the user to access any resource to which the user is authored through SAF.

The following prerequisites must be met to establish a z/OSMF single sign-on environment:

- ▶ All participating z/OSMF servers must be in the same LTPA domain as the primary z/OSMF server.
- ▶ You must have a user ID that must be the same in all SAF user registries. The use of a RACF Remote Sharing Facility (RRSF) ensures that the IDs match. We recommend that your security settings are identical in all participating z/OSMF instances.
- ▶ Use one common CA certificate or sharing CA certificates to enable SSL connections between your z/OSMF instances.
- ▶ LTPA keys are generated when the z/OSMF server starts and there is no defined LTPA key file. This file is encrypted with a randomly generated key and is protected with a user-defined password. The default password is WebAS. We recommend that you change this password on your primary z/OSMF server, stop and start your server to regenerate the LTPA key file, and then, proceed with establishing the single sign-on environment, as shown in Figure 1-18.

The screenshot shows a web-based dialog box titled 'Change LTPA Key' for system 'SC76_001'. At the top, there are tabs for 'Welcome' and 'Systems', with 'Systems' being the active tab. Below the tabs, a breadcrumb trail reads 'Systems > Change LTPA Key'. The main heading is 'Change Security Key for SC76_001'. A paragraph of text explains: 'Specify the LTPA key that is used to protect the LTPA key file. To activate the new key, restart the z/OSMF server on system 'SC76_001'. The systems listed in the Systems field have single sign-on enabled. If you change the LTPA key, re-enable single sign-on for these systems.' Below this, it shows 'Date LTPA credentials last generated (GMT): Sun Sep 27 16:40:13 GMT 2015'. There are two input fields: '* New security key:' and '* Confirm new security key:'. At the bottom, there are 'OK' and 'Cancel' buttons.

Figure 1-18 z/OSMF Single Sign-on support: Change LTPA key

- All servers must share the LTPA key. For z/OSMF, this sharing is done by selecting **z/OSMF Settings** → **Systems** → **Actions** → **Enable Single Sign-on** to synchronize the LTPA key on the primary and secondary z/OSMF servers, as shown in Figure 1-19.

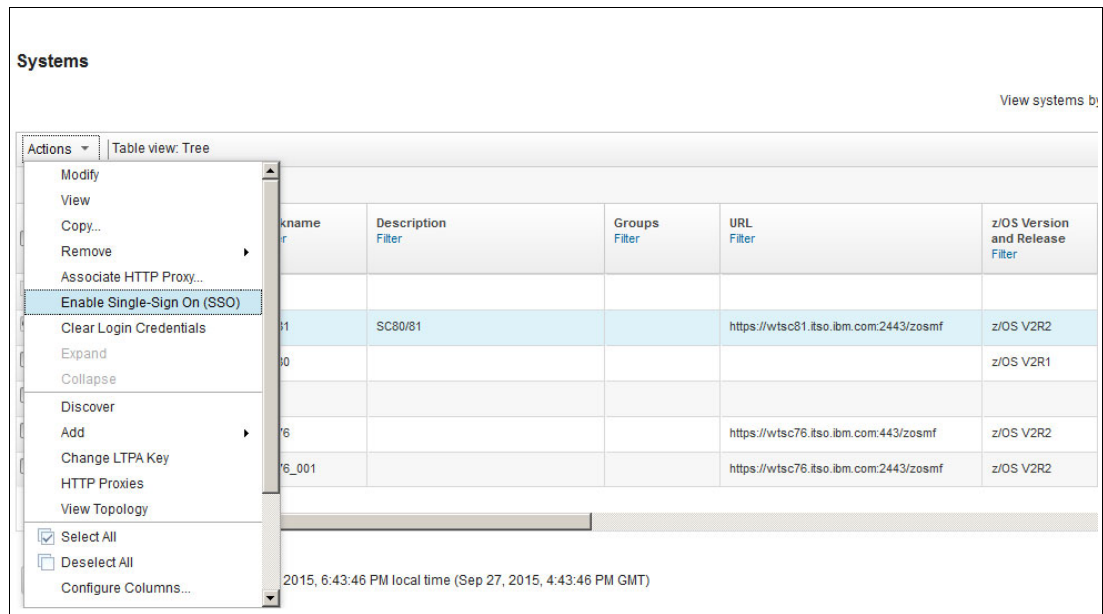


Figure 1-19 z/OSMF enabling single sign-on support

Here, you select one secondary in the system table and click **Enable Single Sign-on**. The selected system must contain its valid z/OSMF URL. Restart the z/OSMF server on the selected system to activate the new LTPA key. The primary z/OSMF server and the selected secondary server then use the same LTPA key.



z/OS Common Event Adapter enhancements

This chapter describes enhancements for the z/OS Common Event Adapter (CEA) in V2R2.

2.1 CEA overview

CEA was first introduced with z/OS V1R9. It enables Common Information Model (CIM) providers to identify, receive, and process the following selected z/OS events:

- ▶ Write to operator (WTO) and Event Notification Facility (ENF)
- ▶ Program-initiated events

CEA is a non-cancelable address space that runs under z/OS. It is automatically started during Master Scheduler initialization. Current users of CEA are z/OSMF users for its Incident Log task, and Base Control Program internal interface (BCPii) for event processing. CEA features its own parmlib member, CEAPRMxx, and can be modified by using operator commands, such as F CEA,D,PARMS.

This command shows you the current parmlib settings.

2.1.1 CEA Time Sharing Option address space control

In z/OS V2R2, CEA is enhanced to control Time Sharing Option (TSO) address spaces that are programmatically started with new parmlib statements. This enhancement is implemented by adding new keywords with their values in the CEAPRMxx parmlib member, as shown in Figure 2-1.

```
TSOASMGR
(
  MAXSESSIONS(50)           <--from 0 to 2000
  MAXSESSPERUSER(10)        <--from 1 to 10
  RECONSESSIONS(0)
  RECONTIME(00:00:00)
)
```

Figure 2-1 CEAPRMxx parmlib member enhancement for TSO address spaces

Use and invocation

Figure 2-2 shows the use of these parameters by running the **MODIFY** command.

```
F CEA,D,PARMS
CEA0023I COMMON EVENT ADAPTER
STATUS: ACTIVE-FULL CLIENTS: 0 INTERNAL: 0
CEA = (00)
SNAPSHOT = Y|N
HLQLONG = hlqlong HLQ = hlq
BRANCH = bbb COUNTRYCODE = ccc
CAPTURE RANGE FOR SLIP DUMPS:
LOGREC = HH:MM:SS LOGRECSUMMARY= HH:MM:SS
OPERLOG = HH:MM:SS
CAPTURE RANGE FOR ABEND DUMPS:
LOGREC = HH:MM:SS LOGRECSUMMARY= HH:MM:SS
OPERLOG = HH:MM:SS
CAPTURE RANGE FOR CONSOLE DUMPS:
LOGREC = HH:MM:SS LOGRECSUMMARY= HH:MM:SS
OPERLOG = HH:MM:SS
VOLSER, 1-4 = volumnxx, volumexx, volumexx
TSOASMGR
RECONSESSIONS = nn RECONTIME = hh:mm:ss
MAXSESSIONS = nnnn MAXSESSPERUSER = nn
```

Figure 2-2 New output of F CEA,D,PARMS operator command

2.1.2 Benefit and value

By using this enhancement, you can control the number of TSO sessions. It is used by the z/OSMF ISPF classic interface.



IBM Health Checker

This chapter describes the IBM Health Checker and includes the following topics:

- ▶ 3.1, “Health Checker overview” on page 28
- ▶ 3.2, “REXX support for persistent data” on page 33

3.1 Health Checker overview

IBM Health Checker for z/OS is a system component that identifies potential problems before they affect the availability of your installation or at worst, cause outages. You can see Health Checker in context with the following components:

- ▶ Predictive Failure Analysis (PFA).
- ▶ Runtime Diagnostics, which is described in Chapter 5, “Runtime diagnostics” on page 43.
- ▶ IBM zAware (IBM z Advanced Workload Analysis Reporter). This component runs in a separate logical partition (LPAR) on an IBM z13™ system and provides near real-time detection of anomalous situations in your z/OS system and sysplex environment. IBM zAware is based on past system behavior and continuous monitoring OPERLOG. It is pattern recognition analytics. It raises awareness of small problems before they affect your environment.

Its target is to reduce the mean time to recovery (MTTR). IBM zAware monitors z/OS OPERLOG messages, including all z/OS console messages, messages from non IBM products, and application generated messages. It uses a 90-day status baseline for initial adjustment and reports in 10-minute intervals, which are updated every two minutes. Results are shown by using the IBM zAware GUI in a browser.

All of these components are designed to circumvent sick but not dead (SBND) failures in your z/OS environment.

IBM introduced Health Checker in z/OS V1R7 and increased the number of checks with nearly every new z/OS release.

Health Checker checks the currently active settings and definitions for a system and compares the values to those values that are suggested by IBM or defined by you.

It produces output in the form of detailed messages that inform you about potential problems and suggested actions to take. In a running z/OS environment, Health Checker and its checks can be managed from System Display and Search Facility (SDSF). In a running z/OS environment, you can use the System Display and Security Facility (SDF) to manage the status of the health checks.

In each z/OS release, you can see new and updated checks, which are described next.

3.1.1 New or updated health checks

New and updated health checks in z/OS V2R2 apply to the following topics:

- ▶ Catalog
- ▶ Communication Server for z/OS
- ▶ Component Trace (CTRACE)
- ▶ Device Manager
- ▶ Hardware Accelerator Manager
- ▶ Integrated Cryptographic Support Facility for z/OS
- ▶ IOS
- ▶ JES2 and JES3
- ▶ PFA
- ▶ RACF
- ▶ System Symbols
- ▶ TSO/E
- ▶ UNIX System Services

- ▶ XCF
- ▶ zFS
- ▶ HTTP server migration

These checks are described next.

Catalog

This check (IBMCATALOG, CATALOG_ATTRIBUTE_CHECK) looks for catalogs with inconsistent share options.

Communication Server for z/OS

The following checks are available:

- ▶ (IBMCS, ZOSMIGV2R1_CS_LEGACYDEVICE): This check looks for TCP/IP profile statements for older device types that are removed in z/OS V2R2.
- ▶ (IBMCS, ZOSMIGV2R2_NEXT_CS_SENDMAILCLIEN): This check looks for use of the sendmail client on z/OS. IBM suggests that you migrate to the Search Results communications server simple mail transfer protocol (CSSMTP) daemon that was introduced in z/OS V1R11.
- ▶ (IBMCS, ZOSMIGV2R2_NEXT_CS_SENDMAILDAEMN): This check looks for the use of the sendmail daemon on z/OS. IBM suggests that you migrate to the CSSMTP daemon that was introduced in z/OS V1R11.
- ▶ (IBMCS, ZOSMIGV2R2_NEXT_CS_SENDMAILMSA): This check looks for the use of the sendmail mail submission agent (MSA) on z/OS. If MSA is needed, IBM suggests that you migrate the functions to another operating system platform that has support for MSA functions. This support can be Linux on IBM z Systems™.
- ▶ (IBMCS, ZOSMIGV2R2_NEXT_CS_SENDMAILMTA): This check looks for the use of the sendmail mail transfer agent (MTA) on z/OS. If MTA is needed, IBM suggests that you migrate the functions to another operating system platform that has support for MTA functions. This support can be Linux on IBM z Systems.
- ▶ (IBMCS, ZOSMIGV2R2_NEXT_CS_SMTPDDAEMON): This check looks for the use of the simple mail transfer protocol daemon (SMTPD) MTA on z/OS. If SMTPD MTA is needed, IBM suggests that you migrate the functions to another operating system platform that has support for MTA functions. This support can be Linux on IBM z Systems.
- ▶ (IBMCS, ZOSMIGV2R2_NEXT_CS_SMTPDMTA): This check looks for the use of the SMTPD on z/OS. IBM suggests that you migrate to the CSSMTP daemon that was introduced in z/OS V1R11 to send mail from the z/OS JES spool. If you use SMTP for purposes other than sending mail from the JES spool, IBM suggests that you migrate the functions to another operating system platform that has support for full email functions. This support can be Linux on IBM z Systems.

Note: Support for the SMTPD, sendmail daemon, MSA function, and MTA function is planned to be withdrawn in the release after the V2R2 IBM z/OS Communications Server.

Component trace (CTRACE)

This check (IBMCTRACE, CTRACE_DEFAULT_OR_MIN) looks for active component traces that are tracing with more than the default or minimum options for an extended amount of time. Tracing with extra options for longer times can cause degradation in system performance.

Device Manager

This check (IBMDMO, DMO_REFUCB) looks for the enablement status of the REFUCB function, which is recommended to maintain VTOC integrity with shared DASD.

Hardware Accelerator Manager

This check (IBMHWAM, HWAM_ZEDC_DEVICE_AVAILABILITY) looks at the current IBM zEnterprise® Data Compression (zEDC) device configurations. The check gives an exception if the available devices are in the same failure domain or if fewer devices than expected are available and active. This check is done to avoid single point of failures.

Integrated Cryptographic Support Facility

The following Integrated Cryptographic Support Facility (ICSF) checks are available:

- ▶ (IBMICSF, ICSFMIG77A1_COPROCESSOR_ACTIVE): This check detects cryptographic coprocessors that do not become active when Integrated Cryptographic Support Facility (ICSF) with FMID HCR77A1 or later is started.
- ▶ (IBMICSF, ICSFMIG77A1_TKDS_OBJECT): This check detects any token data set (TKDS) object that is too large to allow the TKDS to be read into storage during ICSF initialization. ICSF with FMID HCR77A1 introduces a new common key data set (KDS) record format for common cryptographic architecture (CCA) key tokens and PKCS#11 tokens, and objects (public key cryptographic standards), which adds new fields for key utilization and metadata. Because of the size of the new fields, some PKCS#11 objects in the TKDS can cause ICSF to fail to start.
- ▶ (IBMICSF, ICSFMIG77A1_UNSUPPORTED_HW): This check detects if the system is supported by ICSF FMID HCR77A1. This check is necessary because the new ICSF FMID does not support the older IBM zSeries 800 and 900 systems.
- ▶ (IBMICSF, ICSF_KEY_EXPIRATION): This check examines the key validity end date of each record in each active KDS. It checks the record label, checks to determine whether the end date is set, and informs you about records that expire in a defined number of days.

IOS

This check (IBMIOS, IOS_DYNAMIC_ROUTING) identifies any inconsistencies in the dynamic routing support for I/O requests within the storage area network (SAN).

JES2

This check (IBMJES2, JES2_UPGRADE_CKPT_LEVEL_JES2) verifies the status of the JES2 \$ACTIVATE level. It is a renamed and updated version of the previous CHECK(IBMJES2, JES2_Z11_UPGRADE_CK_JES2). It issues exceptions if required preconditions are not met for upgrading to the new JES2 z22 checkpoint level. For example, the z22 mode requires EAV support (3390 extended address volumes).

JES3

The following checks are available:

- ▶ (IBMJES3, JES3_DATASET_INTEGRITY): This check determines whether Decision Server Insights (DSI) or NODSI was specified on the JES3 entries in the program properties table (PPT). IBM suggests that you use DSI to avoid corruption and cold starts for JES3.
- ▶ (IBMJES3, JES3_DOT_POOL_USAGE): This check verifies the use of the JES3 data set output table (DOT) cellpool as a percentage of the pool's total capacity.

- ▶ (IBMJES3, JES3_JET_POOL_USAGE): This check verifies the use of the JES3 job data set entry (JDE) table (JET) cellpool as a percentage of the pool's total capacity. It generates an exception message when the current usage exceeds a threshold that is specified.
- ▶ (IBMJES3, JES3_OST_POOL_USAGE): This check verifies the use of the JES3 output scheduling element (OSE) summary table (OST) cellpool as a percentage of the pool's total capacity. It generates an exception message when the current usage exceeds a specified threshold.
- ▶ (IBMJES3, JES3_SEE_POOL_USAGE): This check verifies the use of the JES3 SYSOUT application programming interface (SAPI) exclusion elements (SEE) cellpool as a percentage of the pool's total capacity.

PFA

This check (IBMPFA, PFA_PRIVATE_STORAGE_EXHAUSTION) examines if there is a potential for private virtual storage to be exhausted by any address space in the future. It replaces any removed slots and frames check. For more information about PFA updates in z/OS V2R2, see “Private storage exhaustion check” on page 38.

RACF

The following checks are available:

- ▶ (IBMRACF, RACF_CSFKEYS_ACTIVE): This check verifies whether the CSFKEYS RACF resource class is active. It is recommended to protect cryptographic keys with profiles in RACF class CSFKEYS.
- ▶ (IBMRACF, RACF_CSFSESV_ACTIVE): This check examines if the CSFSESV RACF resource class is active. The same suggestion applies here as described in “PFA” on page 31.
- ▶ (IBMRACF, RACF_ENCRYPTION_ALGORITHM): This check verifies that a *secure* password authentication algorithm is in effect. If you do not have the new RACF AEF-based password encryption active, you see the following message:
 “IRRH293E KDFAES encryption is not enabled on your system”. The result has no effect on your system.
- ▶ (IBMRACF, RACF_PASSWORD_CONTROLS): This check examines password policies and shows IBM recommendations. The severity of this check is MEDIUM. Consider the following points:
 - Enable mixed-case passwords.
 - Set the invalid password revocation count to three or less.
 - Set the maximum number of days for a valid password or passphrase to 90.
 - Activate the INITSTATS function to enable other options that enhance login security.
- ▶ (IBMRACF, RACF_RRSF_RESOURCES): This check verifies the security attributes of the INMSG/OUTMSG workspace data sets for the RACF remote sharing facility (RRSF) node.
- ▶ (IBMRACF, RACF_SENSITIVE_RESOURCES): This updated check with a severity of HIGH now examines more general resources: ICSF PKDS, CKDS, and TKDS data sets (if present). It also checks the following z/OS UNIX resources:
 - Class FACILITY:
 - BPX.FILEATTR.SHARELIB
 - BPX.JOBNAME
 - BPX.POE
 - BPX.SMF

- BPX.STOR.SWAP
- BPX.UNLIMITED.OUTPUT
- Class UNIXPRIV:
 - SUPERUSER.IPC.RMID
 - SUPERUSER.FILESYS.PFSCTL
 - SUPERUSER.FILESYS.QUIESCE
 - SUPERUSER.FILESYS.VREGISTER
 - SUPERUSER.SETPRIORITY
- Class SURROGAT: BPX.SRV.userid

System symbols

This check (IBMSUP, SUP_SYSTEM_SYMBOL_TABLE_SIZE) examines whether the (used) size of the static system symbol table exceeded a specified threshold. (There is a minor update in V2R2 for an increased maximum symbol table size.)

TSO/E

This check (IBMTSOE, TSOE_OPERSEWAIT_SETTING) examines the OPERSEWAIT setting from the IKJTSOxx parmlib member (where TSO/E commands and programs are defined) if it is explicitly specified. IBM recommends that you not use the setting OPERSEWAIT(ON) because it causes the OPERATOR SEND command by default to be issued with WAIT when nothing is explicitly specified in the SEND command. This issue might lead to resource contention or storage shortages, depending on the available resources and storage. In z/OS V2R2, the default for OPERSEWAIT is switched to OFF.

UNIX System Services

The following checks are available:

- ▶ (IBMUSS, USS_KERNEL_PVTSTG_THRESHOLD): This check monitors the availability of private storage in the kernel that is below the 2 GB bar. If no private storage is available, some UNIX System Services syscalls fail, which can result in a system outage.
- ▶ (IBMUSS, USS_KERNEL_RESOURCES_THRESHOLD): This check monitors the current use of z/OS UNIX System Services kernel resources and dependent items (such as number of threads) when parameter KERNELSTACKS(ABOVE) is active in BPXPRMxx member of parmlib. If your system is running out of kernel resources, it can cause system calls to start failing. Thus, the severity of this check is HIGH.
- ▶ (IBMUSS, USS_KERNEL_STACKS_THRESHOLD): This check monitors the kernel supply of autodata cellpool cells. If KERNELSTACKS(ABOVE) is specified in BPXPRMxx parmlib member, this check is not valid.

XCF

The following checks are available:

- ▶ (IBMXCF, XCF_CF_SCM_UTILIZATION): This check informs an installation when coupling facility storage-class memory (SCM) reaches certain usage thresholds. For each coupling facility in the sysplex, you receive information about its name, the total defined SCM space, and its utilization as a percent.
- ▶ (IBMXCF, XCF_CF_STR_MAXSCM): This check compares the total SCM that is configured to the coupling facility to the sum of the SCM that is eligible to be assigned to allocated coupling facility structures. It is important to avoid over commitment of SCM by a coupling facility. If you do not have structures that use SCM, this check is not applicable.

- ▶ (IBMXCF, XCF_CF_STR_MAXSPACE): This check compares the real storage resources that are available in a coupling facility to the sum of the maximum structure sizes and estimated augmented space values of allocated coupling facility structures, plus the total dump space. It is important to avoid over commitment of real storage by a coupling facility.
- ▶ (IBMXCF, XCF_CF_STR_SCM_UTILIZATION): This check looks for coupling facility structures that exceed one of the specified SCM utilization thresholds.
- ▶ (IBMXCF, XCF_CF_STR_SCM_MAXSIZE): This check scans for the SCM that is available for assignment by a coupling facility to an allocated structure being equal to the SCMMAXSIZE value of a CFRM policy structure definition. If you do not have any structures that use SCM, this check is not applicable.
- ▶ (IBMXCF, XCF_CF_STR_SCM_MINCOUNTS): This check verifies that the number of structure objects that are allocated to a structure meets the required minimum for structures that can be duplexed according to the active CFRM policy. If you do not have any structures that use SCM, this check is not applicable.
- ▶ (IBMXCF, XCF_CF_STR_SCM_AUGMENTED): This check scans for coupling facility structures that have residual in-use augmented real storage space after all structure objects are removed from SCM. Residual augmented space prevents alter processing from dynamically adjusting coupling facility structure storage usage. If you do not have any structures that use SCM, this check is not applicable.

zFS

The following checks are available:

- ▶ (IBMZFS, ZFS_CACHE_REMOVALS): This check looks for user-specified zFS IOEFSPRM configuration options METABACK_CACHE_SIZE, CLIENT_CACHE_SIZE, and TRAN_CACHE_SIZE. Client cache and transaction cache no longer are supported in z/OS V2R2 and later as 64-bit support allows zFS to obtain caches that are above the bar.
- ▶ (IBMZFS, ZFS_VERIFY_CACHESIZE): This updated check scans the settings for IOEFSPRM configuration options: META_CACHE_SIZE, METABACK_CACHE_SIZE and USER_CACHE_SIZE. If you run your system with a small cache size, this size can affect your zFS performance.

HTTP server migration

This check (IBMZMIG, ZOSMIG_HTTP_SERVER_DOMINO_CHECK) examines which HTTP server is in use, if you have one running in your system. The IBM HTTP Server Powered by Domino is being removed in z/OS V2R2. IBM recommends migrating to IBM HTTP Server Powered by Apache. This check is not automatically shipped with z/OS V2R2 or any other authorized program analysis report (APAR). Instead, you must download it from the following website:

http://www.ibm.com/systems/z/os/zos/installation/HTTP_Health_Checker.html

3.2 REXX support for persistent data

z/OS V2R2 Health Checker provides REXX programming language support to allow health checks to store and retrieve persistent data (saved across initial program loads), as is supported for checks written by using High Level Assembler. This new function makes it easier to write health checks.

Health checks are available for the following different sources:

- ▶ Individual IBM components
- ▶ ISVs
- ▶ General users

Each check is intended to focus on one scenario only

Health checks are named (CheckOwner, CheckName). Consider the following points:

- ▶ CheckOwner is typically CompanyName_ComponentName
- ▶ CheckName is typically ComponentName_CheckPurpose

For example: (IBMASM, ASM_PLPA_COMMON_USAGE) is an IBM Auxiliary Storage Manager (ASM) check, which looks at the slot use of the PLPA and common page data sets.

3.2.1 Persistent data

Health Checker maintains a single data set for persistent data that is identified by the HZSPDATA DD statement in the HZSPROC procedure that is used to start the Health Checker address space. It also can be identified by the HZSPDATA statement in a HZSPRMxx parmlib member (available in z/OS V2R1 and higher). The required format is shown in SYS1.SAMPLIB(HZSALLCP).

Services HZSPWRIT and HZSPREAD allow write and read. HZSPWRIT can write for the calling check only. HZSPREAD can read data for its own and other checks. If you plan to access the HZSPDATA, you must define profiles in the RACF class XFACILIT.

Example 3-1 shows how to define a RACF profile read access to HZSPDATA data set for your own check.

Example 3-1 RACF profile definitions for read access to HZSPDATA set

```
RDEFINE XFACILIT HZS.sysname.checkowner.checkname.PDATA UACC(NONE)
PERMIT HZS.sysname.checkowner.checkname.PDATA CLASS(XFACILIT)
      ID(hzspdid) ACCESS(READ)
SETROPTS RACLIST(XFACILIT) REFRESH
```

A RACF profile for the HZSPREAD service has ACCESS(UPDATE) instead.

Each check has up to four sets of its own record or records in HZSPDATA. Health Checker keeps data for the current and prior initial load. For each of those initial loads, a first and a most recent instance is kept.

Writing persistent data in REXX

A new program HZSLPDWR (*Persistent Data Write*) is available. This program is a REXX callable wrapper for the write service HZSPWRIT. HZSLPDWR parameters are passed via REXX variables, such as Health Checker REXX functions. One example is HZSLFMSG, which is the REXX callable wrapper for sending check messages.

HZSLPDWR uses the following new variables:

For Input variables:

- ▶ HZSLPDWR_BUFFER
 - Data to be written, up to 64 K characters/bytes
 - Length is taken from implicit REXX variable length

- ▶ HZS_HANDLE (implicit)

For Output variables:

- ▶ HZSLPDRD_RC, HZSLPDRD_RSN

For any nonzero return code, the system also attempts to Issue HZSFMSG REQUEST=STOP REASON=ERROR DIAG=diag. (For more information, see the Health Checker User's Guide for list of rc, rsn, and diag values.)

- ▶ HZSLPDRD_SYSTEMDIAG

More diagnostic information for nonzero return codes.

An example of how to write persistent data in REXX is shown in Example 3-2.

Example 3-2 Writing persistent data for health checker in REXX

```

/* Language: REXX */
CALL HZSLSTRT
IF HZS_PQE_FUNCTION_CODE="INITRUN" | HZS_PQE_FUNCTION_CODE="RUN"
THEN
DO
  X = 17
  Y = "ABC"
  Z = 31.7
  HZSLPDWR_BUFFER = X,"Y","Z"
  CALL HZSLPDWR
  IF (HZSLPDWR_RC <> 0)
    THEN SAY "HZSLPDWR RSN=" HZSLPDWR_RSN
END
CALL HZSLSTOP
RETURN

```

3.2.2 Reading persistent data in REXX

There is a new program HZSLPDRD (*Persistent Data Read*), which is a REXX callable function to wrapper for read service HZSPREAD. The HZSLPDRD parameters are passed via REXX variables. Its Input and output variables are similar to the HZSPREAD service.

An example of how to read persistent data in REXX is shown in Example 3-3.

Example 3-3 Reading persistent data for health checker in REXX

```

/* Language: REXX */
CALL HZSLSTRT
IF HZS_PQE_FUNCTION_CODE="INITRUN" | ,
  HZS_PQE_FUNCTION_CODE="RUN" THEN
DO
  HZSLPDRD_CHECKOWNER = HZS_PQE_CHECKOWNER /* our own */
  HZSLPDRD_CHECKNAME = HZS_PQE_CHECKNAME /* our own */
  HZSLPDRD_IPL = "CURRENT"
  HZSLPDRD_INSTANCE = "MOSTRECENT"
  /* Default: HZSLPDRD_STARTBYTE = "FIRST_BYTE" */
  /* Default: HZSLPDRD_DATALEN = "MAX" */
  CALL HZSLPDRD
  IF (HZSLPDRD_RC <> 0)
    THEN SAY "HZSLPDRD RSN" HZSLPDRD_RSN

```

```

ELSE
DO
    PARSE VAR HZSLPDRD_BUFFER X","Y","Z
    SAY "X:" X", Y:" Y", Z:" Z", RETIPLTOD:" C2X(HZSLPDRD_RETIPLTOD)
END
END
CALL HZSLSTOP
RETURN

```

3.2.3 Migration checks

Migration checks are health checks that help with the task of identifying the need for migration actions. These checks are shipped inactive and are manually activated on demand, often before an actual migration.

A common naming convention for migration checks is (CheckOwner,ZOSMIGxyz_RestOfCheckName) and ICSF (Integrated Cryptographic Support Facility) that uses ICSFMIG as prefix. The xyz can be used for the following designations:

- ▶ VxRy, for migration actions required for a specific release (“V2R2”)
- ▶ VxRy_NEXT and VxRy_NEXT2, for recommended migrations that become required in one or two releases after VxRy
- ▶ “REC”, for recommended migration actions without a specific release for the foreseeable future
- ▶ “REQ”, for recommended actions that become required soon, but have no release attached to them yet

Examples for migration checks are shown in Example 3-4.

Example 3-4 Migration Checks

```

(IBM CNZ, ZOSMIGV1R13_CNZ_Cons_Oper_Mode)
(IBM ZFS, ZOSMIGREC_ZFS_RM_MULTIFS)

```

The benefit for this technique is that the growing number of REXX checks can use a consistent and easy interface for maintaining persistent data.



IBM predictive failure analysis

This chapter describes the IBM predictive failure analysis (PFA) enhancements in z/OS V2R2 and includes the following topics:

- ▶ 4.1, “PFA overview” on page 38
- ▶ 4.2, “PFA summary of changes” on page 38

4.1 PFA overview

Software detected system failures are categorized into one of the following types:

- ▶ **Masked failures:** These failures are software detected system failures, which are detected and corrected by the software component.
- ▶ **Hard failures:** This failure occurs when the software fails completely; for example, when an operator stops a process.
- ▶ **Failures that are caused by abnormal behavior:** These failures are unexpected or unusual situations that cause the software component to not provide the requested service. In combination with events that often do not generate failures, secondary effects can occur that might eventually result in a system or sysplex outage.

The idea behind PFA is to predict potential problems that might arise in your z/OS environment. These potential problems are not hard failures; instead, these problems are soft failures that can be categorized into the following areas:

- ▶ Exhaustion of shared resources
- ▶ Recurring failures that are caused by damage to critical control structures
- ▶ Serialization problems, such as classic deadlocks and priority inversions
- ▶ Unexpected state transitions

These soft failures can lead to situations that are called *sick but not dead* (SBND). These situations have the following characteristics:

- ▶ It is difficult for components to be detected internally.
- ▶ They are probabilistic, but not deterministic.

Situations that arise in this area cause 20% of the problems. Because of their long duration, these situations generate 80% of the business impact.

PFA was originally made available as a small program enhancement (SPE) in z/OS V1R10 with the first two checks, and then officially in z/OS V1R11 with the next two checks. These checks reviewed the following factors:

- ▶ Common storage usage
- ▶ LOGREC arrival rate
- ▶ Virtual storage usage
- ▶ Message arrival rate

In z/OS V2R2, PFA is improved to ease the set-up and installation processes by completing more verification when it starts.

4.2 PFA summary of changes

This section describes the new functions that were introduced in IBM z/OS V2R2.

4.2.1 Private storage exhaustion check

In z/OS V2R2, PFA monitors several ranges of private area virtual storage for multiple address spaces and warns you when one or more address spaces exceed criteria that can indicate eventual private area virtual storage exhaustion. PFA is constructed largely in Java, and most of the CPU that is used by PFA can run on IBM System z® Integrated Information Processor (zIIP) specialty engines.

In this release, address spaces can be included or excluded by using the new INCLUDED_JOBS file or the EXCLUDED_JOBS file, which you can specify to monitor critical jobs or those persistent batch jobs for processing message queues in your installation. Data that is collected for Jobs supports dynamic checks for the following components:

- ▶ PFA_PRIVATE_STORAGE_EXHAUSTION
- ▶ PFA_JES_SPOOL_USAGE
- ▶ PFA_MESSAGE_ARRIVAL_RATE
- ▶ PFA_SMF_ARRIVAL_RATE
- ▶ PFA_ENQUEUE_REQUEST_RATE

PFA can be dynamically updated by using z/OS console commands for the following private storage exhaustion and JES spool usage checks:

- ▶ F PFA,UPDATE,CHECK(PFA_P*),INCLUDED_JOBS
- ▶ F PFA,UPDATE,CHECK(PFA_J*)

The PFA_PRIVATE_STORAGE_EXHAUSTION check detects future exhaustion of private storage that is under 2 GB in six storage locations within the following individual address spaces:

- ▶ Private user region: USER
- ▶ Private authorized area: AUTH
- ▶ Private user and private authorized: BELOW the line
- ▶ Extended private user region: EUSER
- ▶ Extended private authorized area: EAUTH
- ▶ Extended private user and extended private authorized: ABOVE the line

Figure 4-1 shows the different virtual storage locations in z/OS that are detected by this check.

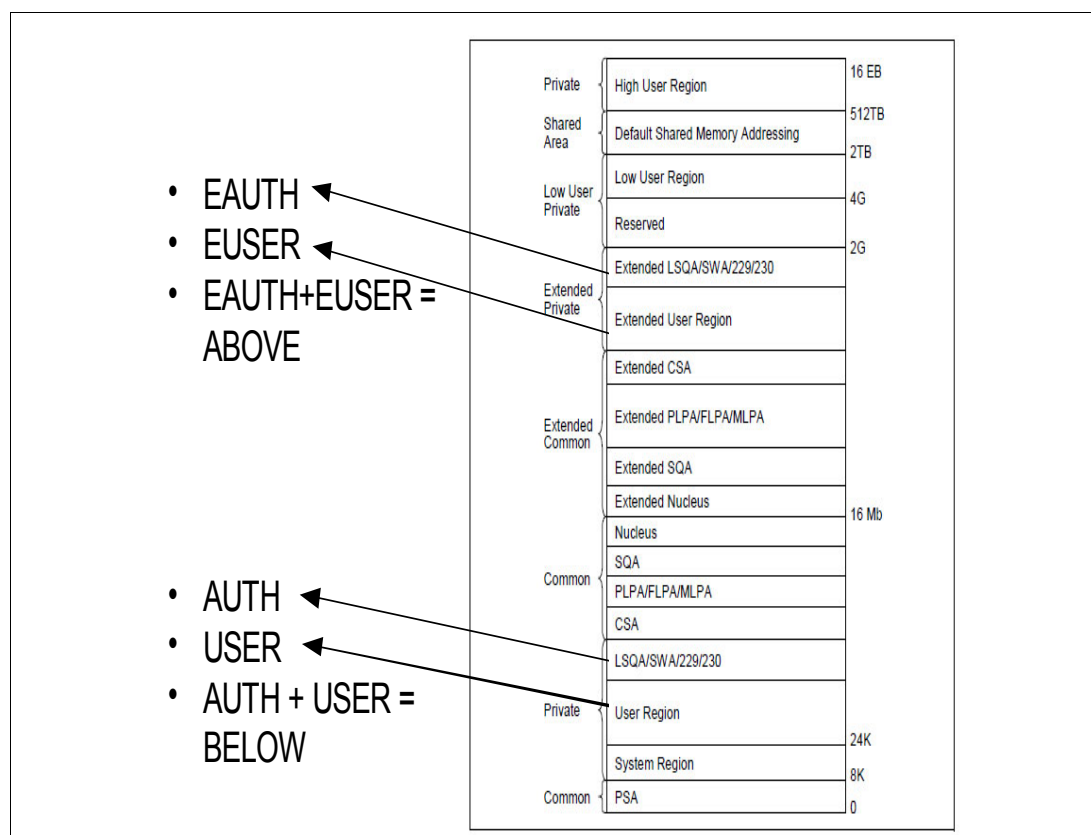


Figure 4-1 Virtual storage locations detected by PFA check PFA_PRIVATE_STORAGE_EXHAUSTION

The following components are shown in Figure 4-1:

- ▶ USER: User region in the private area
- ▶ EUSER: User region in the extended private area
- ▶ AUTH: LSQA, SWA, subpools 229, and 230 in the private area
- ▶ EAUTH: LSQA, SWA, subpools 229, and 230 in the extended private area
- ▶ ABOVE: The extended user private area above 16 M (the sum of EUSER+EAUTH)
- ▶ BELOW: The user private area below 16 M (the sum of USER+AUTH)

This check does not detect exhaustion that is caused by the following factors:

- ▶ Fragmentation
- ▶ Fast increases of usage that are on a machine-time scale or even faster than one collection interval

PFA uses dynamic severity, which means that as *time to exhaustion* gets closer, the severity of the PFA exception increases. It is used for PFA_COMMON_STORAGE_USAGE and PFA_PRIVATE_STORAGE_EXHAUSTION checks only.

As with the other PFA checks, this check is added to the Health Checker when PFA is started. If you want to see all current values for this check, enter the following command:

```
F PFA,DISPLAY,CHECK(PFA_P*),DETAIL
```

The result of this command is shown in Figure 4-2.

```
F PFA,DISPLAY,CHECK(PFA_P*),DETAIL
AIR018I 17:33:21 PFA CHECK DETAIL 461
CHECK NAME: PFA_PRIVATE_STORAGE_EXHAUSTION
  ACTIVE                : YES
TOTAL COLLECTION COUNT   : 1733
SUCCESSFUL COLLECTION COUNT : 1733
LAST COLLECTION TIME     : 08/17/2015 17:31:42
LAST SUCCESSFUL COLLECTION TIME: 08/17/2015 17:31:42
NEXT COLLECTION TIME     : 08/17/2015 17:36:42
TOTAL MODEL COUNT        : 12
SUCCESSFUL MODEL COUNT   : 12
LAST MODEL TIME          : 08/17/2015 10:05:57
LAST SUCCESSFUL MODEL TIME : 08/17/2015 10:05:57
NEXT MODEL TIME          : 08/17/2015 22:05:57
CHECK SPECIFIC PARAMETERS:
  COLLECTINT             : 5
  MODELINT               : 720
  COLLECTINACTIVE        : 1=ON
  DEBUG                  : 0=OFF
  EXCDIR DAYS            : 90
  FORCEMODEL              : NO
  COLL%                  : 20
  COLLUPTIME             : 180
  MOD%                   : 40
  COMP%                  : 100
  E_HIGH                 : 180
  E_MED                  : 300
  E_LOW                  : MAX
  E_NONE                 : UNUSED
```

Figure 4-2 Output of the PFA display command

Figure 4-2 also shows the following defaults for the time to exhaustion:

- ▶ E_HIGH(180): If time to exhaustion is predicted to be 0 - 180 minutes from now, a critical eventual action WTO is issued.
- ▶ E_MED(300): If time to exhaustion is predicted to be from more than E_HIGH minutes to 300 minutes from now, an eventual action write to operator (WTO) is issued.
- ▶ E_LOW(MAX): If time to exhaustion is predicted to be from more than E_MED minutes to the expiration of the prediction, an informational WTO is issued.
- ▶ E_NONE(UNUSED): A value of 0 or UNUSED for the number of minutes indicates that this dynamic severity is not used.

This information is also used for the PFA_COMMON_STORAGE_USAGE check.

Benefit and value

This check improves system availability by providing information that you can use before storage exhaustion if the rate of storage consumption is excessive.



Runtime diagnostics

This chapter describes the enhancements that were implemented for runtime diagnostics (RTD) that are running in z/OS V2R2.

RTD is a z/OS component that helps to find and remove soft failures that might lead to sick but not dead (SBND) situations.

This chapter includes the following topics:

- ▶ 5.1, “RTD overview” on page 44
- ▶ 5.2, “Health-based routing integration with runtime diagnostics” on page 46

5.1 RTD overview

RTD was originally introduced in z/OS V1R12. It analyzes SBND systems quickly and searches for evidence of soft failures. For more information about soft failures, see “PFA overview” on page 38. Soft failures can be of the following areas:

- ▶ Component issues
- ▶ Global resource contention
- ▶ Important address space execution issues

You can use RTD when your operations staff report a problem on the system. The benefit of RTD is that it provides a timely, comprehensive analysis at a critical time without the need for a storage dump. This advantage can save you time.

You can use RTD to quickly analyze an ailing system for the following types of problems:

- ▶ Component problems that are identified as critical messages in OPERLOG
- ▶ ENQ, GRS latch contention for system address spaces, and z/OS UINX file system contention
- ▶ Address spaces with high CPU usage
- ▶ Address spaces that appear to be in a task control block (TCB) enabled loop
- ▶ Local lock conditions
- ▶ JES2 health exceptions
- ▶ Server address space health exceptions

With that information, you can take the next step, including the following tasks:

- ▶ Cancel the relevant jobs
- ▶ Further investigate the class of resources, or a single address space by using a monitor, such as IBM RMF™ or Omegamon XE for z/OS.

Use the following z/OS command to start RTD from your console or SDSF:

```
S HZR,SUB=MSTR
```

You can then start analyzing your system by entering the following command:

```
F HZR,ANALYZE
```

When you enter the analyze command, a report displays, as shown in Figure 5-1.

```
F HZR,ANALYZE
HZR0200I RUNTIME DIAGNOSTICS RESULT 319
SUMMARY: SUCCESS
REQ: 001 TARGET SYSTEM: SC81      HOME: SC81      2015/08/18 - 13:40:11
INTERVAL: 60 MINUTES
EVENTS:
FOUND: 04 - PRIORITIES: HIGH:02 MED:02 LOW:00
TYPES: CF:01 DUMPS:02 ENQ:01
-----
EVENT 01: HIGH - ENQ          - SYSTEM: SC81      2015/08/18 - 13:40:11
ENQ WAITER - ASID:0035 - JOBNAME:HZSPROC - SYSTEM:SC81
ENQ BLOCKER - ASID:0014 - JOBNAME:HZSPROC - SYSTEM:SC81
QNAME: SYSDSN
RNAME: SYS1.SC81.HZSPDATA
ERROR: ADDRESS SPACES MIGHT BE IN ENQ CONTENTION.
ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE BLOCKING JOBS AND
ACTION: ASIDS.
-----
EVENT 02: HIGH - CF          - SYSTEM: SC81      2015/08/18 - 12:42:41
IXC585E STRUCTURE HZS_HEALTHCHKLOG IN COUPLING FACILITY CF8B,
PHYSICAL STRUCTURE VERSION CF61F249 A0D1E082,
IS AT OR ABOVE STRUCTURE FULL MONITORING THRESHOLD OF 80%:
      SPACE USAGE      IN-USE      TOTAL      %
ENTRIES:              1645        1954      84
ERROR: INDICATED STRUCTURE IS APPROACHING FULL MONITORING THRESHOLD.
ACTION: D XCF,STR,STRNAME=strname TO GET STRUCTURE INFORMATION.
ACTION: INCREASE STRUCTURE SIZE OR TAKE ACTION AGAINST APPLICATION.
-----
EVENT 03: MED - DUMPS        - SYSTEM: SC81      2015/08/18 - 13:05:10
IEA799I AUTOMATIC ALLOCATION OF SVC DUMP DATASET FAILED
DUMPID=018 REQUESTED BY JOB (CONSOLE )
DYNALLOC FAILED RETURN CODE=04 ERROR RSN CODE=970C INFO RSN CODE=0000
SMS RSN CODE=4379
ERROR: THE SYSTEM WAS UNABLE TO ALLOCATE A DUMP DATA SET FOR A DUMP.
ACTION: D D TO VIEW ALLOCATION STATUS. DD ADD,VOL=volser TO ADD DUMP
ACTION: RESOURCES.
-----
EVENT 04: MED - DUMPS        - SYSTEM: SC81      2015/08/18 - 13:05:20
IEA799I AUTOMATIC ALLOCATION OF SVC DUMP DATASET FAILED
DUMPID=019 REQUESTED BY JOB (HSIBMGR )
DYNALLOC FAILED RETURN CODE=04 ERROR RSN CODE=970C INFO RSN CODE=0000
SMS RSN CODE=4379
ERROR: THE SYSTEM WAS UNABLE TO ALLOCATE A DUMP DATA SET FOR A DUMP.
ACTION: D D TO VIEW ALLOCATION STATUS. DD ADD,VOL=volser TO ADD DUMP
ACTION: RESOURCES.
-----
```

Figure 5-1 Output of the RTD analyze command

5.2 Health-based routing integration with runtime diagnostics

In z/OS V2R2, health-based routing is an enhancement to Workload Manager (WLM) dynamic workload routing. The focus here is to further reduce the effect that is caused by middleware or transaction manager server health issues.

WLM provides a health service that is called IWM4HLTH to enable multiple callers to report on a server's health. The server identifies itself and can provide reasons for its health ratings.

When you run the `F HZR,ANALYZE` command, RTD starts a new query service that is called IWM4QHLT. This service obtains server health states. The information is then used for diagnostic and serviceability purposes.

If any servers show a current health value that is less than 100, a `SERVERHEALTH` event is returned to PFA, and PFA starts RTD for health checks that can indicate that the metric is too low. The event is included in the predictive failure analysis (PFA) check exception report. The health indicator is a number that shows how well a server is performing. It can be an integer number of 0 - 100.

Benefits of these new functions are improved routing recommendations and diagnostic reporting about server health states.



Subsystem initialization and management

This chapter describes improvements in subsystem initialization and management. The introductory level information about z/OS subsystems helps you to understand the z/OS V2R2 modifications that are described.

This chapter includes the following topics:

- ▶ 6.1, “Subsystem overview” on page 48
- ▶ 6.2, “Subsystem initialization problems” on page 49
- ▶ 6.3, “Benefit and value” on page 53

6.1 Subsystem overview

The word *subsystem* has many meanings in data processing. Here, we are defining a precise entity that is named *z/OS subsystems*.

z/OS subsystem is a set of programs that provides several services when requested. They can be in a specific address space, private area, or in a common area. A z/OS subsystem has the following properties:

- It is started by using the macro IEFSSREQ, also called the *SSI interface*, as shown in Figure 6-1.

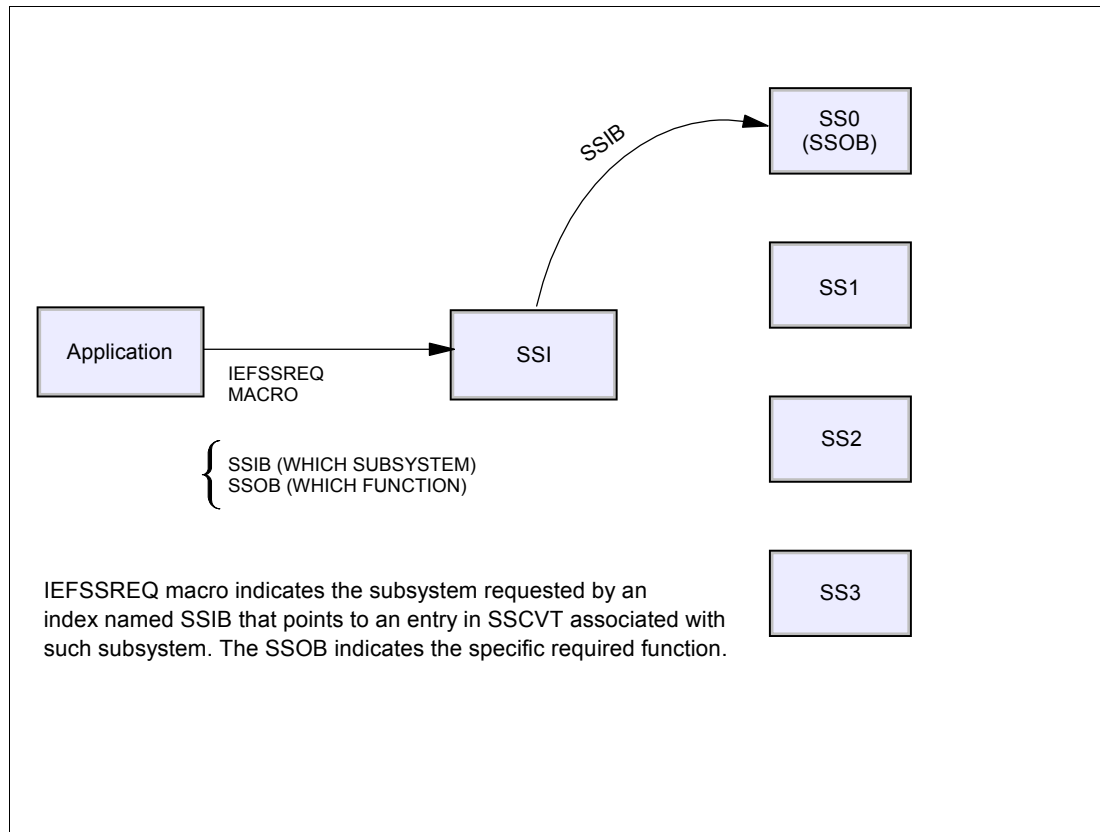


Figure 6-1 SSI macro and interface control blocks

- Every subsystem is automatically informed about all commands and messages that are passing through Multi Console Support (MCS) consoles.
- Defined as a subsystem to z/OS in a z/OS control block named SSCVT, through the following components:
 - IFSSNxx parmlib member at initial program load with the following contents:
 - Subsystem name and whether you want it to start automatically
 - Name of the subsystem initialization exit routine (INITRTN) to be given control during master scheduler initialization after IPL/NIP processing
 - Input parameter string to be passed to the subsystem initialization routine
 - An authorized program that issues the IEFSSI macro, any time after initial load
 - System console SETSSI ADD command any time after initial load

Subsystems can be dynamic or static. A dynamic subsystem can be defined by any one of the three methods that were described in the previous list. The static subsystems can be defined by starting the macro IEFFSREQ.

The following types of subsystems are available:

- ▶ MSTR, the master scheduler z/OS component that is activated after NIP processing. It is necessary for running z/OS.
- ▶ Primary, such as JES2 or JES3. It is necessary for running z/OS.
- ▶ Secondary, or the other subsystems, such as IBM CICS®, DB2, and RACF.

6.2 Subsystem initialization problems

Before z/OS V2R2, when you define a new subsystem to your z/OS, the subsystem name is reserved, regardless of whether it might be successfully started.

If an error occurs, such as at the INITRTN routine, you cannot redefine the subsystem or create another subsystem that uses the same name. If the subsystem does not provide another way to complete the initialization processing, the subsystem is left in limbo.

For example, a subsystem that is installed with incorrect, not installed, or non-APF authorized INITRTN routine can fail activation because of these errors. Even if it can be disabled or deactivated, the subsystem name is still reserved and cannot be reused. This issue often requires zapping or initially loading the production systems to fix the problem. These remedies are risky and can lead to system outages. Another way to bypass the problem is to define the subsystem with a different name, which is inconvenient.

z/OS V2R2 modifications

Starting with z/OS V2R2, multiple changes were introduced with which you can delete and redefine subsystems.

The new features that are described in this chapter are default to z/OS V2R2, and no installation is required. RACF changes are required to support subsystems. The following new features are described next:

- ▶ INITRTN pre-processing
- ▶ SETSSI DELETE command
- ▶ EVENTRTN exit routine
- ▶ DISPLAY SSI command

6.2.1 INITRTN pre-processing

The initialization processing was updated on z/OS V2R2 to prevent INITRTN errors.

Implementation and usage

When a subsystem with INITRTN is defined, the z/OS subsystem initialization function checks to determine whether the INITRTN routine is installed in a library and that it is APF-authorized. If any errors are found, the error message that is shown in Example 6-1 on page 50 is displayed.

Example 6-1 Subsystem initialization failure for INITRTN error

```
IEFJ027I subsystem INITIALIZATION ROUTINE yyyyyyy NOT  
FOUND FOR subsystem xxxx
```

This message replaces the two error messages, as shown in Example 6-2.

Example 6-2 INITRTN error messages replaced

```
IEE859I subsystem xxxx NOT INITIALIZED - yyyyyyy NOT  
FOUND  
IEFJ004I subsystem xxxx NOT INITIALIZED - yyyyyyy  
NOT FOUND
```

If you have an automation process to track these messages, make sure to update them to the new message ID and text.

If errors are identified, the subsystem is not defined and no manual recovery actions are necessary. You can redefine the subsystem (with the same name) by using SETSSI ADD or another method appropriate for the subsystem, when applicable.

The changes in the INITRTN initialization routine process are default to z/OS V2R2, and no implementation steps are required.

Note: Only subsystems with INITRTN parameters are affected by this change. Also, ABENDs in INITRTN or errors in the INITPARM are not affected by this change.

6.2.2 SETSSI DELETE

Before z/OS V2R2, subsystems that were defined to the z/OS were not deleted by using regular commands. With z/OS V2R2, a new SETSSI DELETE command is introduced to allow logical deletion of subsystems. This ability is useful for the subsystems in error, which allows the installation to reuse the subsystem names.

Implementation and usage

The SETSSI DELETE command is available by default in z/OS V2R2. There is a RACF security profile in class OPERCMDS to control access to the DELETE and ADD commands. Figure 6-2 and Figure 6-3 show how to define these RACF profiles.

```
RDEFINE OPERCMDS MVS.SETSSI.DELETE.ssnm UACC(NONE)  
PERMIT MVS.SETSSI.DELETE.ssnm CLASS(OPERCMDS)  
ID(userid) ACCESS(READ)  
SETROPTS RACLIST(OPERCMDS) REFRESH
```

Figure 6-2 RACF profile definition for the SETSSI DELETE command in class OPERCMDS

```
RDEFINE OPERCMDS MVS.SETSSI.ADD.ssnm UACC(NONE)  
PERMIT MVS.SETSSI.ADD.ssnm CLASS(OPERCMDS)  
ID(userid) ACCESS(READ)  
SETROPTS RACLIST(OPERCMDS) REFRESH
```

Figure 6-3 RACF profile definition for the SETSSI ADD command in class OPERCMDS

The `ssnm` portion of the RACF profile can include the following characters:

- ▶ **Exact characters**

If your subsystem name includes lowercase characters, use uppercase characters to define the resource name. For a subsystem with the name “abcd”, define a profile that is called `IBM MVS™.SETSSI.*.ABCD` in RACF.

- ▶ **Special characters:**

If your subsystem name includes special characters, such as “*”, “&”, or “%”, use the underscore character “_” in the resource name instead. For a subsystem name “SUB%”, define a profile that is called `MVS.SETSSI.*.SUB_` in RACF.

- ▶ **Blank spaces:**

If your subsystem name includes embedded blanks, use the underscore character “_” in the resource name instead. Trailing blanks do not use the underscore character. For a subsystem name “A BC”, define a profile that is called `MVS.SETSSI.*.A_BC`.

- ▶ **Asterisk (*):**

If you must use other unsupported characters or if you want to define generic profiles, use the asterisk.

Although the **SETSSI DELETE** command performs a logical deletion of the subsystem, its routines that are processing at the time of the command are not ended and any memory that is associated with the subsystem (such as load modules or control blocks) is not freed.

Command syntax

To delete a subsystem, you must issue the **SETSSI DELETE** command, along with the subsystem name and the **FORCE** option. The subsystem name can be enclosed in single quotation marks if it contains nonstandard characters, or the hexadecimal name can be used. The hexadecimal requires all 4 bytes to be specified. If needed, specify trailing blanks ('40'X).

In Example 6-3, you can see how to issue the command and its resulting message.

Example 6-3 SETSSI DELETE command syntax

```
SETSSI DELETE,SUBNAME=I9N2,FORCE
IEFJ022I SETSSI DELETE COMMAND FOR subsystem I9N2 COMPLETED
SUCCESSFULLY
```

The **FORCE** option is required when you enter the command. Otherwise, message “ASA100I SYNTAX ERROR” is shown.

It is not required that the subsystem be a dynamic subsystem to be deleted. Individual subsystems can have their own requirements regarding deletion. You might be required to take more steps to delete a specific subsystem. Some subsystems might not support deletion at all or they might support deletion but cannot be dynamically added.

For example, you cannot delete the MSTR subsystem or the primary subsystem (JES2/JES3). When a subsystem is deleted, the following tasks are performed to complete the deletion process:

- ▶ The SSI attempts to deactivate the subsystem so that its function routines no longer receive control.
- ▶ Various internal control blocks and queues are updated to reflect that the subsystem no longer exists.
- ▶ If necessary, create a dummy subsystem.

The subsystem name is !DMY, and it is an inactive subsystem. The !DMY subsystem is created only when you delete the last subsystem that was defined. This requirement is necessary to maintain serialization on SSI control blocks.

- ▶ An SSCVT (which represents a subsystem) is created to replace the subsystem to be deleted.

The subsystem name is !DEL, and it is an inactive subsystem. A !DEL subsystem is created for every subsystem that is deleted. You can have multiple !DEL subsystems defined. The reason is that SSCVT is an indexed table and its entries are accessed through indexes.

- ▶ The !DEL SSCVT is swapped onto the SSCVT chain, which replaces the SSCVT for the subsystem that is deleted.
- ▶ If the subsystem has an EVENTRTN (new in z/OS V2R2), it is called to notify the subsystem that it was deleted.

Attention: SETSSI DELETE is a *force* command. Removing an active subsystem can have far-reaching side effects. Consider it to be a command that is used as a last resort.

6.2.3 EVENTRTN exit routine

In addition to initialization error handling, the new parameter that is named EVENTRTN can be added to the IEFSSI macro. This parameter informs the address of an exit routine. This routine gains control for notifying the subsystem when a specific event occurs, which is called an event notification facility (ENF). An ENF is a z/OS component that notifies users (listeners) about the occurrence of certain events. As of this writing, only the DELETE event is supported. Here, a DELETE event means the issue of the SETSSI DELETE command. However, the EVENTRTN is designed so that other events can be supported in the future.

The EVENTRTN option is available for dynamic subsystems only, and it is defined through the IEFSSI macro and not by the console command SETSSI ADD. Example 6-4 shows a sample use of the EVENTRTN keyword.

Example 6-4 Use of EVENTRTN keyword

```
IEFSSI SUBNAME=subname,  
      REQUEST=OPTIONS,  
      EVENTRTN=exitname
```

The environment for the EVENTRTN is similar to the INITRTN; that is, key 0, supervisor state. However, the EVENTRTN must be AMODE 31. The DELETE event occurs *after* the subsystem is deleted, which means that the subsystem no longer exists.

If you write an EVENTRTN, make sure that you check the event type before performing any processing and ignore any event types that you do not recognize.

6.2.4 DISPLAY SSI command

The DISPLAY SSI command is enhanced to display the subsystem name in hexadecimal, and any EVENTRTN that is associated with the subsystem. Example 6-5 shows the output from the DISPLAY SSI,ALL command.

Example 6-5 Sample DISPLAY SSI,ALL output

```
IEFJ100I 15.25.44 SSI DISPLAY
SUBSYS=SDB1      HEX=E2C4C2F1
      DYNAMIC=YES  STATUS=ACTIVE      COMMANDS=REJECT
FUNC=236 237 239 240
EVENTRTN=YES
```

6.3 Benefit and value

The benefit of these new functions is a faster and easier recovery from subsystem-related errors.

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the z/OS V2R2 updates. Some of the publications that are referenced in this list might be available in softcopy only:

- ▶ *z/OS V2R2: JES2, JES3, and SDSF*, SG24-8287
- ▶ *z/OS V2R2: Security*, SG24-8288
- ▶ *z/OS V2R2: Storage Management and Utilities*, SG24-8289
- ▶ *z/OS V2R2: Availability Management*, SG24-8290
- ▶ *z/OS V2R2: Performance*, SG24-8292
- ▶ *z/OS V2R2: Operations*, SG24-8305
- ▶ *z/OS V2R2: Diagnostics*, SG24-8306
- ▶ *z/OS V2R2: Sysplex*, SG24-8307
- ▶ *z/OS V2R2: UNIX System Services* SG24-8310
- ▶ *z/OS V2R2: User Interfaces*, SG24-8311
- ▶ *z/OS V2R2: ServerPac*, SG24-8500

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft, and other materials, at the following website:

ibm.com/redbooks

Other publications

The following publications are also relevant as further information sources:

- ▶ *IBM z/OSMF V2R2 Configuration Guide*, SC27 8419
- ▶ *IBM z/OSMF V2R2 Programming Guide*, SC27-8420
- ▶ *z/OS MVS Initialization and Tuning Guide*, SA23-1379

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services





SG24-8290-00

ISBN 0738441295

Printed in U.S.A.

Get connected

